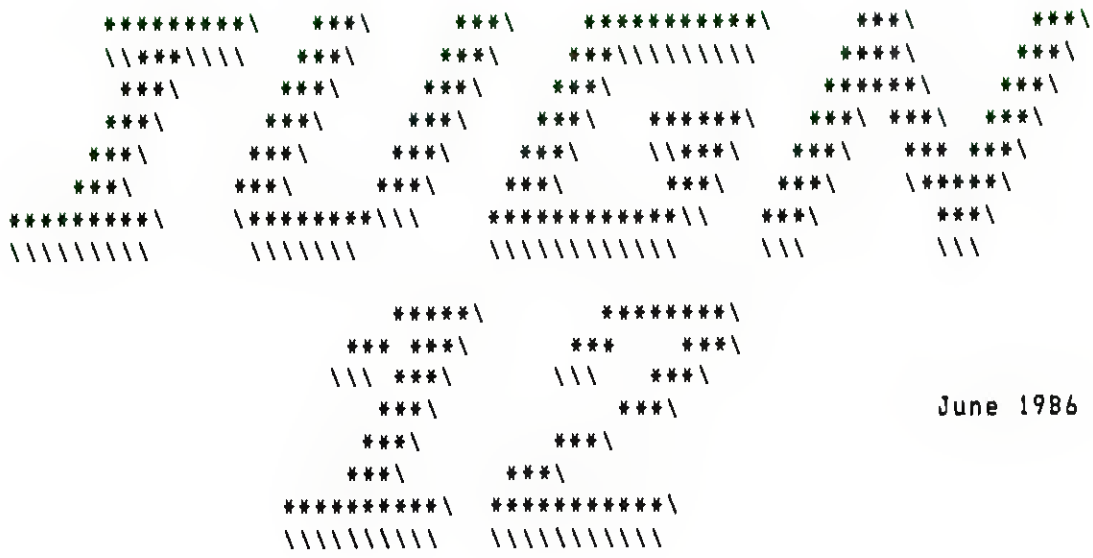
Interaktion



June 1986

DATA PROTECTION ACT 1985:
Details of the members of the IUGN are held on computer file. Each member may view, alter or destroy any data held on him/her within that file. To obtain a copy of your file please send a stamped addressed envelope to the Editor. Your subsequent wishes regarding that data will be honoured without question.

NOTES

UGN-11 was 48 pages times 750 copies, equals 36,000 pages, carefully produced and distributed. "Phew", as David said when the last stamp was licked.

THE INTERAK BULLETIN BOARD

Tom Evans runs "TOMS BULLETIN BOARD" and this is free to all IUGN members.

Data line is 01-573-8822 at 300 BAUD
Speech line is 01-561-2639 for help and advice

I must thank Simon Waller who has sent a copy of the

H064180 CMOS 8-BIT MICROPROCESSOR USER'S MANUAL.

This is for insertion into the book library, I will be passing it on to the library as soon as possible. Is this our Z80 replacement at last?. Simon will reveal all in a forthcoming issue. If you can't wait till then write to Simon at Flat 2, Maritime Court, 58 Foundation Street, Ipswitch, IP4 1BN. I am sure he will be glad to pass on some of his knowledge of this very powerfull 512k byte address space, 6Mhz, 64k I/O space chip. Please enclose a stamped addressed envelope for his conveniance.

Would somebody like to be a disk librarian?. No pay, long hours, usually evening work, no thanks, great sense of satisfaction and high personality growth. Applications to David of Greenbank also send a short resume of your requirements, (Charges for copying, disk costs, etc) to me for publication in the next newsletter.

ASM64/32
When using ASM64 or ASM32 be careful where you load your object code to, as a member recently experienced a fault which when investigated showed that he was LOADing to address 2000H. And as ASM64/32 uses store from 1000H-31FFH for its own purposes, and from 3200H onwards for your source file it caused major file corruptions.

I advise using :-

```
ORG  8000H
LOAD 8000H
```

This gives 3200H-7FFFH for the source and 8000H-08FFFH for the object. If your object exceeds 8k you should be moving to disk based assemblers.
If your final program is to run at say 1000H, then test it at 8000H and when it works use :-

```
ORG  1000H
LOAD 8000H
```

Dump it to tape with ZYMONs S 8000 XXXX CR finally loading it back in to run in its final home.

The reason for the M command in ASM is for you to know how big your source file is both for planning the LOAD space and for tape saving with ZYMON.

A map of ASM64 shows
1000H-1FFFH ASM64/32 code
2000H-31FFH symbol tables, vars and stack.
3200H-xxxx user source file

TAPES

Many members are kindly sending in XTAL BASIC programs, for use in the newsletter, on tape. This is of real value in ensuring accuracy of typsetting and I am very gratefull to members who do this. However I would like to request that the following be considered when making up these tapes.

1. Please save the program using the ASC directive. I.e "MYPROG.ASC". This saves the program in ASCII format and makes my task of getting the data from tape to disk much easier.

2. Use 1200 BAUD as a maximum save rate. 2400 BAUD is difficult to transfer.

3. Please put one program on the tape. If you send two programs put the second one on the reverse side.

4. Remember that others have to type your programs into their computer. To you as the programmer this may be ok : -

```
1890 FORT=1TO255:OUT&C0,0:OUT&C1,T:PRINT@42,7,">"T :NEXTT
```

But to me, when trying to edit the file and the reader when trying to copy it into his machine, this is much simpler : -

```
1890 FOR T=1 TO 255: OUT &C0,0: OUT &C1,T: PRINT @42,7,">" T: NEXT T
```

All I have done is to space it out a bit.

5. Please keep line lengths to 75 characters. If you don't then I have to chop the lines to fit on the page. This can cause problems for readers trying to count characters into their computer.
If the program requires lines longer than 75 characters then you must of course include them. But try to minimise their use.

The above rules are to help me. They can be broken if the program has special needs. I will in the future add a space after every ";" to improve readability.

Bob Eldridge

Pi = 0011.0001000100000011011101010101001001000010000100000010011001000011000 0

O B I T U A R Y, W O L F S C H R O E D E R

It is with great sadness that I have to report the recent death of Wolfgang (Wolf) Schroeder at the age of 62.

He collapsed without warning walking to work on 13th February 1986. I am not certain of the details but it appears he suffered a heart attack triggered by the bitterly cold weather that day, and he died a few minutes later.

Although he was born and educated in Germany he spent the majority of his life in England, married to his English wife Winifred, to whom we extend our deepest sympathy for her loss, also to his daughters Stephanie and Bernice, and his grandchildren, Belinda, Ian and Veronique.

He was a lecturer in Operational Research for Cranfield Institute of Technology, and had worked with computers (IBM mainframes downwards) from their earliest days. A particular speciality, which was of great relevance to Interak users was his expertise with disk operating systems, especially CP/M. Wolf's influence was immense on the design of the DMON series of disk monitors and the present implementation of CP/M version 2.2 on Interak. Before disks were generally available for home use, he adapted CP/M so that it would work on a high speed tape system of his own design, and he worked for a long time with just a single low-capacity 5.25" disk drive, so he had the combined experience of working with large highly expensive disk systems of enormous capacities professionally, and with small systems at home.

Wolf was one of the first users of Interak, even though it was then not known as Interak of course. In 1978 he purchased a product of ours called the SC/MP skeleton pack, which was a set of chips to build a small computer (128 bytes) based on the National Semiconductors SC/MP microprocessor and the companion RAM-I/O chip. In those days it was thought very unlikely that any individual would ever be able to build and understand a computer based on highly advanced chips such as the Z80.

He wrote a number of text-books on astronomy (in German and in translation), and even though they were written in the fifties they are still in print. One very interesting use he had for his Interak computer was the collection of data from a set up of detectors for some private research he was doing on the frequency and distribution of electrical storms in pretty well the whole of England. He never took time off completely from academic pursuits, but his favourite recreation was walking in the Lake District and Wales.

He never spoke much about his life in Germany, but I know he was in the German Army for a time. He was multilingual, and spoke scrupulously correct English. He and I shared a mutual interest in English orthography, and I am not ashamed to say I learned a great deal on this subject from him, and I am very flattered that a man who could and did earn thousands of pounds as a consultant was willing to spend so much time with me and the users, passing on the benefit of his wide experience.

He was one of the select band of Interak users who, although they perhaps

don't realise it, have a direct influence on the design and continuing development of the Interak system. Although I cannot of course run the Interak business simply to suit the needs and wishes of a tiny minority, I nevertheless listen to everyone who offers an opinion; for example "my" design of floppy disk interface was altered out of all recognition by valuable suggestions and changes from Bob Eldridge, Wolf Schroeder and others, and similarly without Tom Evans there would be no serial interface design for modems and communication with bulletin boards. Likewise the new Z80A controller card the "SBC-1" was designed by us specifically to suit one purpose Wolf had in his consultancy work, but it will be of great use to many others. Recent extensive correspondence before his death will have a direct bearing on the design of my current project, a universal EPROM programmer, provisionally entitled PGM-1.

For several weeks after he died, and even now, I couldn't get out of the habit of waiting for his regular phone calls so that we could exchange news of how our individual projects were progressing, and offer mutual encouragement and support if we hit some impossible snag, usually hardware snags in his case, software in mine.

Of course Wolf left many jobs unfinished. He had promised to give detailed notes on how the disk routines in the DMON monitors could be called by users, and he intended to share a Pascal compiler he had designed which would allow fast programs to be written on an Interak disk system, and then down loaded into an EPROM to stand alone on the single board SBC-1 card. Users however need not worry too much about support for Interak systems which contain elements from Wolf Schroeder; Interak seems to have an individual life of its own, and there is always someone who will step in and solve problems like this which arise.

One of my greatest sadnesses is that I never met Wolf in person. Like many other of my Interak friends, I get to know each one very well by telephone and letter, but very rarely meet them face to face. (I think it is often a shock for them when they meet me!) However by all accounts it appears that Wolf was tall, a pipe smoker, distinguished looking, always immaculately turned out, usually with a bow tie, and thought highly of by his professor, staff, and students at Cranfield. Mrs Schroeder has written to me and described his funeral and how he left the college he loved with dignity on his last journey. With his flowers his family placed a quotation from one of Wolf's first books on astronomy.

Although the world goes on without him, for many of us it will never be the same place it was before. He will continue to be sadly missed.

David M Parkins.

Note:
When Wolf sent documents such as the description of the disk monitor program for publication he had particularly asked that his name and address appear on them so that any users would write to him if they needed help or had any comments to make. Obviously that arrangement can no longer apply and I would be grateful if all users would erase his address from all documents so that Mrs Schroeder will not suffer any further distress by receiving enquiries intended for Wolf.

PROGRAMAMMABLE CHARACTER DEFINITION
VERSION 35
FOR XTAL BASIC
BY F.R.JOHNSON

From. Mr F.R.Johnson, 32 Langdon Road, Folkestone, Kent, CT19 4HX.

Dear Mr Eldridge,

Please find enclosed a program I have written which other members may find useful. It stemmed directly from my success with the programmable high resolution character generator card, which initially I had some problems with. It now functions perfectly as a daughter board plugged into the VDU-2K, and so I have written the program to more easily design new characters and symbols.

All the instructions are 'on board', so I'll say no more.

P.S. If you know of other members who have had difficulty getting the high res' card to work properly, I can send them details or I could write a short article for the newsletter if you feel there would be sufficient appeal. Please let me know.

Yours sincerely
F.R. Johnson

(EO: Thanks for your program. I am sure that an article on the hi-res card would be well received by all of us, I look forward to reading it).

```
 10 REM*******************************************************
 20 REM
 30 REM Programmable Character Definition. Version 35.
 40 REM
 50 REM Written by F.R.Johnson.
 60 REM
 70 REM*******************************************************
 80 REM
 90 GOSUB 3550: REM Initialise.
100 GOSUB 3980: REM Introduction.
110 GOSUB 4390: REM Set up initial screen.
120 REM
130 FOR WHILE=0 TO 1: REM          Has anyone added
140 GOSUB 210: REM Main routine.   a WHILE/WEND loop
150 NEXT WHILE: REM                to XTAL Basic yet?
160 CLS
170 END
180 REM
190 REM*******************************************************
200 REM
210 REM Main routine
220 REM
230 E$=INCH$
240 IF E$="?" OR E$="/" THEN GOSUB 780: REM Instructions.
250 IF E$="Q" THEN WHILE=1: RETURN: REM Quit.
260 IF E$="S" THEN CHAR(X-1,Y-1)=1: REM Store set pixel.
270 IF E$="R" THEN CHAR(X-1,Y-1)=0: REM Store reset pixel.
280 IF E$="C" THEN GOSUB 640: REM Clear characters.
```

```
290 IF E$="V" THEN GOSUB 440: REM View characters.
300 IF $E$="M" THEN GOSUB 440: GOSUB 1150: REM Display memory.
310 IF E$="H" THEN X=X+1: REM Cursor right.
320 IF E$="G" THEN X=X-1: REM   "    left.
330 IF E$="Y" THEN Y=Y-1: REM   "    up.
340 IF E$="B" THEN Y=Y+1: REM   "    down.
350 IF X<1 THEN X=1: REM      Stay
360 IF X>XE THEN X=XE: REM    inside
370 IF Y<1 THEN Y=1: REM      matrix
380 IF Y>YE THEN Y=YE: REM    area.
390 IF CHAR(X-1,Y-1)=0 THEN PRINT@X,Y," ";BS$;: ELSE PRINT@X,Y,D$;BS$;:
    REM Print set pixel or cursor.
400 WHILE=0: RETURN: REM Do it again.
410 REM
420 REM*************************************************************
430 REM
440 REM Set up char' RAM & display char's.
450 REM (All 6 char's are stored regardless of block size).
460 REM
470 PRINT@30,9,"Memory being set up, please wait
480 FOR R=0 TO 10 STEP 10: REM Character row.
490 FOR C=0 TO 16 STEP 8: REM Character column.
500 FOR V=0 TO 9: REM Pixel vertical position.
510 FOR H=0 TO 7: REM   "  horizontal   "
520 IF CHAR(H+C,V+R)=1 THEN ROW=ROW+2^(7-H)
530 REM ^ Check piWxel array & convert each row to HEX.
540 NEXT H
550 POKE RAM+V+2*C+R*4.8,ROW: REM Normal character.
560 POKE RAM+&60+V+2*C+R*4.8,&FF-ROW: REM Inverse character.
570 ROW=0
580 NEXT V,C,R
590 PRINT@30,9,SPC(33): REM Clear message.
600 RETURN
610 REM
620 REM*************************************************************
630 REM
640 REM Clear matrix area.
650 REM
660 X=1: Y=1
670 FOR A=1 TO 24
680 FOR B=1 TO 20
690 PRINT@ A,B," ";BS$;
700 CHAR(A-1,B-1)=0
710 NEXT B,A
720 FOR CL=&F800 TO &F859: POKE CL,0: NEXT CL: REM   Restore
730 FOR CL=&F860 TO &F8B9: POKE CL,&FF: NEXT CL: REM memory.
740 RETURN
750 REM
760 REM*************************************************************
770 REM
780 REM Instructions.
790 REM
800 FOR ST=12 TO 22: SCOREEN$(ST-12)=LEFT$(SCRN$(ST),28): NEXT ST
810 REM ^ Store screen contents.
820 GOSUB 1060: REM Clear bottom of screen.
```

```
830 PRINT@0,12,"The following keys are used in this program: -"
840 PRINT@6,14,"Cursor"
850 PRINT@8,15,"up"
860 PRINT@8,16,"Y"
870 PRINT@1,17,"left G   H right"
880 PRINT@8,18,"8"
890 PRINT@7,19,"down"
900 PRINT@22,14,"Q to Quit the program.
910 PRINT@22,15,"C to Clear all pixels in matrix.
920 PRINT@22,16,"S to Set the pixel at the cursor position.
930 PRINT@22,17,"R to Reset ' ' ' ' ' '
940 PRINT@22,18,"V to View the character.
950 PRINT@22,19,"M to Memorise & display the contents.
960 PRINT@5,21,"(A second ? clears these instructions)";
970 PAUSE$=INCH$
980 IF PAUSE$<>"?" AND PAUSE$<>"/" THEN 970
990 GOSUB 1060: REM Clear instructions.
1000 FOR ST=12 TO 22: PRINT@0,ST,SCREEN$(ST-12);: NEXT ST
1010 REM ^ Restore previous screen contents.
1020 RETURN
1030 REM
1040 REM*********************************************************
1050 REM
1060 REM Clear lines 12 to 22.
1070 REM
1080 FOR A=12 TO 22
1090 PRINT@0,A,SPC(64);
1100 NEXT A
1110 RETURN
1120 REM
1130 REM*********************************************************
1140 REM
1150 REM Display memory contents
1160 REM
1170 PRINT@30,9,"Press 'N' for normal character.
1180 PRINT@30,10,"Press 'I' for inverse character.";
1190 QUERY$=INCH$
1200 IF QUERY$="N" THEN REV=0: ELSE REV=&60
1210 GOSUB 1270: GOSUB 1360: GOSUB 1450: GOSUB 3440
1220 REM ^Save char: Clear screen: Print & save: Restore matrix^
1230 RETURN
1240 REM
1250 REM*********************************************************
1260 REM
1270 REM Save matrix contents.
1280 REM
1290 FOR A=0 TO 22
1300 SCREEN$(A)=LEFT$(SCRN$(A),28)
1310 NEXT A
1320 RETURN
1330 REM
1340 REM*********************************************************
1350 REM
1360 REM Clear screen for memory display.
1370 REM
```

```
1380 FOR A=0 TO 21
1390 PRINT@0,A,SPC(26)
1400 NEXT A
1410 RETURN
1420 REM
1430 REM*************************************************************
1440 REM
1450 REM Print memory contents in matrix area.
1460 REM
1470 PRINT@30,9,SPC(32): PRINT@30,10,SPC(32)
1480 ON 8K GOSUB 1580,1640,1700,1760,1820,1880
1490 REM ^ Block  A    B    C    D    E    F
1500 PRINT@30,9,"Do you wish to save it (Y/N)";
1510 IF INCH$="Y" THEN GOSUB 2510: ELSE RETURN
1520 RETURN
1530 REM
1540 REM*************************************************************
1550 REM
1560 REM Call appropiate subroutines for memory contents.
1570 REM
1580 IF REV=0 THEN PRINT@0,0,"CHR$(&80)": ELSE PRINT@0,0,"CHR$(&86)"
1590 GOSUB 1980: REM Block A.
1600 RETURN
1610 REM
1620 REM*********************************
1630 REM
1640 IF REV=0 THEN PRINT@0,0,"CHR$(&80) and CHR$(&81)":
     ELSE PRINT@0,0,P"CHR$(&86) and CHR$(&87)"
1650 GOSUB 1590: GOSUB 2080: REM Block B.
1660 RETURN
1670 REM
1680 REM*********************************
1690 REM
1700 IF REV=0 THEN PRINT@2,0,"CHR$(&80) to CHR$(&82)":
     ELSE PRINT@2,0,"CHR$(&86) to CHR$(&88)"
1710 GOSUB 1650: GOSUB 2170: REM Block C.
1720 RETURN
1730 REM
1740 REM*********************************
1750 REM
1760 IF REV=0 THEN PRINT@0,0,"CHR$(&80) and CHR$(&83)":
     ELSE PRINT@0,0,"CHR$(&86) and CHR$(&89)"
1770 GOSUB 1590: GOSUB 2260: REM Block D.
1780 RETURN
1790 REM
1800 REM*********************************
1810 REM
1820 IF REV=0 THEN PRINT@0,0,"CHR$(&80,&81,&83,&84)":
     ELSE PRINT@0,0,"CHR$(&86,&87,&89,&8A)"
1830 GOSUB 1770: GOSUB 2080: GOSUB 2350: REM Block E.
1840 RETURN
1850 REM
1860 REM*********************************
1870 REM
1880 IF REV=0 THEN PRINT@2,0,"CHR$(&80) to CHR$(&85)":
```

```
     ELSE PRINT@2,0,"CHR$(&86) to CHR$(&88)"
1890 GOSUB 1830: GOSUB 2170: GOSUB 2440: REM Block F.
1900 RETURN
1910 REM
1920 REM************************************************************
1930 REM
1940 REM 6 subroutines to print memory contents in matrix area
1950 REM
1960 REM Top left block.
1970 REM
1980 FOR A=0 TO 9
1990 PRINT@1,A+2,HEX$(&F800+A+REV);"=";FN RHP$(&F800+A+REV)
2000 REM ^Position^     ^MeMmory^       ^Peek^   ^Contents^
2010 NEXT A
2020 RETURN
2030 REM
2040 REM*********************************
2050 REM
2060 REM Top middle block.
2070 REM
2080 FOR A=0 TO 9
2090 PRINT@9,A+2,HEX$(&F810+A+REV);"=";FN RHP$(&F810+A+REV)
2100 NEXT A
2110 RETURN
2120 REM
2130 REM*********************************
2140 REM
2150 REM Top right block.
2160 REM
2170 FOR A=0 TO 9
2180 PRINT@17,A+2,HEX$(&F820+A+REV);"=";FN RHP$(&F820+A+REV)
2190 NEXT A
2200 RETURN
2210 REM
2220 REM*********************************
2230 REM
2240 REM Bottom left block.
2250 REM
2260 FOR A=0 TO 9
2270 PRINT@1,A+13,HEX$(&F1830+A+REV);"=";FN RHP$(&F830+A+REV)
2280 NEXT A
2290 RETURN
2300 REM
2310 REM*********************************
2320 REM
2330 REM Bottom middle block.
2340 REM
2350 FOR A=0 TO 9
2360 PRINT@9,A+13,HEX$(&F840+A+REV);"=";FN RHP$(&F840+A+REV)
2370 NEXT A
2380 RETURN
2390 REM
2400 REM*********************************
2410 REM
2420 REM Bottom right block.
```

```
2430 REM
2440 FOR A=0 TO 9
2450 PRINT@17,A+13,HEX$(&F850+A+REV);"=";FN RHP$(&F850+A+REV)
2460 NEXT A
2470 RETURN
2480 REM
2490 REM*************************************************
2500 REM
2510 REM Save characters to memory.
2520 REM
2530 PRINT@30,9,SPC(33);
2540 PRINT@30,9,"Address? (&F8C0 to &FFF0)";
2550 INPUT" ";MEM$
2560 FAIL=0: GOSUB 3180: REM Check for valid area of RAM.
2570 IF FAIL=1 THEN 2530
2580 PRINT@30,9,"Block saved in CHR$'s as below: - "
2590 ON BK GOSUB 2710,2770,2840,2910,2980,3070
2600 REM ^ Block  A    B    C    D    E    F
2610 PRINT@30,15,"Press a key to return.";
2620 PAUSE=INCH
2630 FOR CL=11 TO 15 STEP2: PRINT@30,CL,SPC(22): NEXT
2640 RETURN
2650 REM
2660 REM*************************************************
2680 REM 6 subroutines to poke characters to memory,
2690 REM corresponding to the 6 block sizes A to F.
2700 REM
2710 FOR M=0 TO 9: POKE MEM+M,PEEK(&F800+REV+M): NEXT M
2720 PRINT@30,11,FN MEM$(MEM): REM Print chosen CHR$ No.
2730 RETURN
2740 REM
2750 REM*********************************
2770 GOSUB 2710
2780 FOR M=0 TO 9: POKE MEM+M+&10,PEEK(&F810+REV+M): NEXT M
2790 PRINT@34,11,FN MEM$(MEM+&10)
2800 RETURN
2810 REM
2820 REM*********************************
2840 GOSUB 2770
2850 FOR M=0 TO 9: POKE MEM+M+&20,PEEK(&F820+REV+M): NEXT M
2860 PRINT@38,11,FN MEM$(MEM+&20)
2870 RETURN
2880 REM
2890 REM*********************************
2910 GOSUB 2710
2920 FOR M=0 TO 9: POKE MEM+M+&10,PEEK(&F830+REV+M): NEXT M
2930 PRINT@30,13,FN MEM$(MEM+&10)
2940 RETURN
2950 REM
2960 REM*********************************
2980 GOSUB 2770
2990 FOR M=0 TO 9: POKE MEM+M+&20,PEEK(&F830+REV+M): NEXT M
3000 FOR M=0 TO 9: POKE M&EM+M+&30,PEEK(&F840+REV+M): NEXT M
3010 PRINT@30,13,FN MEM$(MEM+&20)
3020 PRINT@34,13,FN MEM$(MEM+&30)
```

```
3030 RETURN
3040 REM
3050 REM*******************************
3070 GOSUB 2B40
3080 FOR M=0 TO 9: POKE MEM+M+&30,PEEK(&F830+REV+M): NEXT M
3090 FOR M=0 TO 9: POKE MEM+M+&40,PEEK(&F840+REV+M): NEXT M
3100 FOR M=0 TO 9: POKE MEM+M+&50,PEEK(&F850+REV+M): NEXT M
3110 PRINT@30,13,FN MEM$(MEM+&30)
3120 PRINT@34,13,FN MEM$(MEM+&40)
3130 PRINT@38,13,FN MEM$(MEM+&50)
3140 RETURN
3150 REM
3160 REM************************************************************
3180 REM Check for illegal RAM area.
3190 REM
3200 IF LEFT$(MEM$,1)<>"&"THEN PRINT@30,9,"Hexadecimal only - press a
     key  ";: P=INCH: FAIL=1: RETURN
3210 MEM=VAL(MEM$)
3220 IF MEM<&F8C0 OR MEM>&FFF0 THEN PRINT@30,9,"Out of range - press a key
          ";: P=INCH: FAIL=1: RETURN
3230 IF MEM MOD &10<>0 THEN PRINT@30,9,"Must end in a '0' - press a key";:
     P=INCH: FAIL=1: RETURN
3240 IF PEEK(MEM)<>&FF THEN PRINT@30,9,"Characters are here - press a key";:
     P=INCH: FAIL=1: RETURN
3250 IF BLOCK$="B" AND MEM>&FFE0 THEN GOSUB 3350: RETURN
3260 IF BLOCK$="C" AND MEM>&FFD0 THEN GOSUB 3350: RETURN
3270 IF BLOCK$="D" AND MEM>&FFC0 THEN GOSUB 3350: RETURN
3280 IF BLOCK$="E" AND MEM>&FFB0 THEN GOSUB 3350: RETURN
3290 IF BLOCK$="F" AND MEM>&FFA0 THEN GOSUB 3350
3300 REM  ^ Check for sufficient room in RAM ^
3310 RETURN
3320 REM
3330 REM************************************************************
3350 REM Print 'No room' message.
3360 REM
3370 PRINT@30,9,"No RAM left!! - press a key    ";
33B0 P=INCH
3390 FAIL=1
3400 RETURN
3410 REM
3420 REM************************************************************
3440 REM Restore matrix contents.
3450 REM
3460 PRINT@30,9,SPC(33)
3470 FOR A=0 TO 22
3480 PRINT@0,A,SCREEN$(A)
3490 NEXT A
3500 FOR REV=93 TO 100: SET REV,33: SET REV,3B: NEXT
3510 FOR REV=33 TO 38: SET 93,REV: SET 100,REV: NEXT
3520 REM ^ Redraw inverse border. ^
3530 RETURN
3540 REM
3550 REM************************************************************
3570 REM Initialisation
3580 REM
```

```
3590 CLS
3600 PRINT@9,10,"Initialisation taking place - please wait";
3610 IOM5,0: REM Turn off leading space.
3620 DEF FN RHP$(W)=RIGHT$(HEX$(PEEK(W)),2)
3630 REM ^ Print contents of memory in HEX. ^
3640 DEF FN MEM$(Z)="&"+RIGHT$(HEX$((Z-&F000)/&10),2)
3650 REM ^ Convert save address to CHR$ number.^
3660 DIM CHAR(23,19),SCREEN$(22)
3670 FOR R=&F800 TO &F859: POKE R,0: NEXT: REM   Set up defined
3680 FOR R=&F860 TO &FFFF: POKE R,&FF: NEXT: REM character RAM.
3690 FOR R=&F8C0 TO &F97F: READ Z$: POKE R,VAL("&"+Z$): NEXT R
3700 FOR R=&F980 TO &F98F: POKE R,0: NEXT
3710 FOR R=&F990 TO &F98F: READ Z$: POKE R,VAL("&"+Z$): NEXT R
3720 REM ^ Read data for demonstration characters.^
3730 FRJ$=CHR$(&8C)+CHR$(&8D)+CHR$(&8E)+CHR$(13)+CHR$(10)+CHR$(&8F)+
     CHR$(&90)+CHR$(&91): REM Personal logo.
3740 IUG$=CHR$(&92)+CHR$(&93)+CHR$(&94)+CHR$(13)+CHR$(10)+CHR$(&95)+
     CHR$(&96)+CHR$(&97): REM Interaction User Group logo.
3750 X=1: Y=1: D$=CHR$(&7F): B$=CHR$(255): BS$=CHR$(8): RAM=&F800
3760 NO$=" 012345670123456701234567"
3770 DATA FF,C0,C0,C0,C0,FF,C6,C6,C7,C7,0,0,0,0,0,0
3780 DATA FF,1,1,1,1,E1,31,31,E1,81,0,0,0,0,0,0
3790 DATA FF,80,80,80,80,80,80,80,80,80,0,0,0,0,0,0
3800 DATA C6,6,6,0,0,C,6,3,1,0,0,0,0,0,0,0
3810 DATA C1,61,31,1,1,1,3,6,FC,0,0,0,0,0,0,0
3820 DATA 80,80,80,80,80,80,0,0,0,0,0,0,0,0,0,0
3830 DATA 3F,60,C0,C0,C0,C0,C0,C0,C0,C0,0,0,0,0,0,0
3840 DATA FF,0,3C,18,18,18,18,3C,0,3,0,0,0,0,0,0
3850 DATA FC,6,3,0,0,0,0,0,0,FF,0,0,0,0,0,0
3860 DATA C0,C1,C1,C1,C1,C1,C1,C0,60,3F,0,0,0,0,0,0
3870 DATA 0,81,81,81,81,81,81,FF,0,FF,0,0,0,0,0,0
3880 DATA 3,83,83,83,83,83,83,3,6,FC,0,0,0,0,0,0
3890 DATA FF,81,81,81,81,81,81,81,81,FF,0,0,0,0,0,0
3900 DATA FF,FF,C3,C3,C3,C3,C3,C3,FF,FF,0,0,0,0,0,0
3910 DATA FF,FF,FF,E7,E7,E7,E7,FF,FF,FF,0,0,0,0,0,0
3920 REM ^ Last 6 items in each line are dummy data. ^
3930 REM ^ Only the 1st 10 bytes of each 16 are used.^
3940 RETURN
3950 REM
3960 REM*******************************************************
3980 REM Introduction
3990 REM
4000 CLS
4010 PRINT@17,0,"CHARACTER DEFINITION PROGRAM
4020 PRINT@17,1,MUL$("=",28)
4030 PRINT: PRINT"  This program allows you to program your own characters
     and
4040 PRINT"symbols, either singly or in blocks.You may print the character
4050 PRINT"(and it's inverse form),at any time to see the results, display
4060 PRINT"the memory contents and save up to 115 characters in the new 2K"
4070 PRINT"RAM  from  &F8C0 to &FFFF.  These can then be called as  CHR$'s"
4080 PRINT"from programs in the range CHR$(&8C) to CHR$(&FF)."
4090 PRINT: PRINT"      [ CHR$(&80) to CHR$(&88) are used by the program ]"
4100 PRINT: PRINT"You can, for example, design your own logo's, such as: -
4110 PRINT: PRINT FRJ$;"     (that's the programmers initials)"
```

```
4120 PRINT: PRINT IUG$;"     (a possible candidate for the user group logo?)"
4130 PRINT@9,21,"(a different cursor, perhaps - Press 'C' to continue)";
4140 REM Blinking cursor routine follows.
4150 FOR C=&99 TO &9B: PRINT@1,21,CHR$(C);: GOSUB 45B0: NEXT
4160 PRINT@1,21,CHR$(&9C);: GOSUB 45B0
4170 FOR C=&9B TO &99 STEP-1: PRINT@1,21,CHR$(C);
4180 GOSUB 45B0: NEXT C: GOSUB 45B0
4190 IF KBD$<>"C" THEN 4150
4200 CLS
4210 PRINT@0,3,"You may design one character at a time or up to a block of
     six"
4220 PRINT: PRINT SPC(10);"These are the block sizes to choose from:-
4230 PRINT: PRINT SPC(12);B$;"     ";B$;B$;"     ";B$;B$;B$;"     ";
     B$;"     ";B$;B$;"     ";fB$;B$;B$
4240 PRINT SPC(33);B$;"     ";B$;B$;"     ";B$;B$;B$
4250 PRINT: PRINT SPC(12);"A     B       C       D     E       F
4260 PRINT: PRINT SPC(11);"Please enter the block size (A thru F) ";
4270 BLOCK$=INCH$
4280 IF ASC(BLOCK$)<65 OR ASC(BLOCK$)>70 THEN 4270
4290 IF BLOCK$="A" THEN XE=8: YE=10: BK=1: REM  Set
4300 IF BLOCK$="B" THEN XE=16: YE=10: BK=2: REM matrix
4310 IF BLOCK$="C" THEN XE=24: YE=10: BK=3: REM boundaries
4320 IF BLOCK$="D" THEN XE=8: YE=20: BK=4: REM  & store
4330 IF BLOCK$="E" THEN XE=16: YE=20: BK=5: REM block
4340 IF BLOCK$="F" THEN XE=24: YE=20: BK=6: REM size.
4350 RETURN
4360 REM
4370 REM*************************************************************
4390 REM Set up screen.
4400 REM
4410 CLS
4420 PRINT@0,0,NO$
4430 FOR B=1 TO 10: PRINT@0,B,B-1;SPC(23);B-1: PRINT@0,B+10,B-1;SPC(23);B-1:
     NEXT
4440 PRINT@0,21,NO$
4450 PRINT@30,0,"(Press ?_ for instructions)";
4460 PRINT@31,2,"Normal        Inverse
4470 PRINT@30,3," &B0-&B5       &B6-&BB
4480 PRINT@32,5," ";: FOR CH=&B0 TO &B2: PRINT CHR$(CH);: NEXT:
     PRINT SPC(11);: FOR CH=&B6 TO &BB: PRINT CHR$(CH);: NEXT
4490 PRINT@32,6," ";: FOR CH=&B3 TO/ &B5: PRINT CHR$(CH);: NEXT:
     PRINT SPC(11);: FOR CH=&B9 TO &BB: PRINT CHR$(CH);: NEXT
4500 FOR REV=93 TO 100: SET REV,33: SET REV,3B: NEXT
4510 FOR REV=33 TO 3B: SET 93,REV: SET 100,REV: NEXT
4520 REM ^ Print normal + inverse characters & inverse border.
4530 PRINT@1,1," ";BS$;: REM Put cursor in top left corner.
4540 RETURN
4550 REM
4560 REM*************************************************************
45B0 REM Delay
4590 REM
4600 FOR TIME=1 TO 100: NEXT
4610 RETURN
4630 REM*************************************************************
```

RECTIFIER CIRCUIT PERFORMANCE - SOME NEW FORMULAS
BY A.G.BOGLE

From. A.G.Bogle, 22 Brighton Road, Auckland, New Zealand.

Dear Sir,

It is possible that members of the User Group may find a use for the enclosed program, which I have checked fairly carefully and in which I feel I can repose some confidence. Its aim is to take all (?) guesswork out of the design of DC Power supplies. I suppose there must always be some reservations:

1) In the case of supply to a voltage-regulator, I decided that the minimum drop it required was 2 volts (see line 200) whereas I began to belive that some may well require more.

2) The reason why I have called for a not-too-small Ipk/I in the unregulated case (see lines 1290 & 1510) is that the lower one makes it, by increasing the series resistance, the higher the no-load voltage becomes.

3) Diode Irms is not the same as transformer secondary current, which depends on the diode connection and can readily be derived.

I do think that rectifier circuits designed with the use of the appropriate parts of this program will be more efficient and reliable than those designed "off the top of the head".

I am
Yours truly
A.G. Bogle

This program is based upon the paper of the above title published in Proc I.E.E. Vol.124 p.1127, by A.G.Bogle.

```
  5 PRINT "PERFORMANCE OF RECTIFIER CIRCUIT SUPPLYING REGULATOR OR RESISTIVE
    LOAD": PRINT
 10 INPUT "MAINS FREQUENCY IN HZ" F: PRINT
 20 PRINT "SPECIFY RECTIFIER LOAD: REG OR RES": A$ = INPUT$
 30 IF MID$(A$,1,3)="REG" GOTO 60
 40 IF MID$(A$,1,3)="RES" GOTO 1200
 50 GOTO 20
 60 PRINT: INPUT "REGULATOR OUTPUT VOLTAGE" V1,"LOAD CURRENT IN AMPS" I:
    PRINT: INPUT "RATIO OF DIODE PEAK CURRENT TO LOAD CURRENT" N: PRINT
 70 INPUT "SPECIFY PK-PK RIPPLE VOLTAGE AS PERCENTAGE OF MEAN RECTIFIER
    OUTPUT VOLTAGE" R: PRINT
 80 PRINT "SPECIFY DIODE SERIES IMPEDANCE - L OR R": A$=INPUT$
 90 IF MID$(A$,1,1)="R" GOTO 120
100 IF MID$(A$,1,1)="L" GOTO 700
110 GOTO 80
120 PRINT "SPECIFY H-W OR F-W": A$=INPUT$
130 IF MID$(A$,1,1)="H" GOTO 160
140 IF MID$(A$,1,1)="F" GOTO 360
150 GOTO 120
160 PRINT "H-W RECTIFIER WITH SERIES RESISTANCE"
170 B1=563.7: B2=14: B3=1.5: B4=2.5: B5=2.41: B6=1.41: B9=6.2B3: D=225
180 A1=R/B1
190 Y=B2/N/N/N: GOSUB 390: X=0.6*Y3
200 V3=1.1*(V1+2)
210 GOSUB 520: E2=V3/(1-B3*X3)/1.414/(1-A1*B4)
```

```
220 C2=1E+6*I/89/F/E2/A1
230 Z2=82*E2/I/N/N/N
240 PRINT "Emin =",E2: INPUT "SPECIFY E" E1
250 PRINT "Cmin =",C2: INPUT "SPECIFY C in microF" C1
260 PRINT "Rmin =",Z2: INPUT "SPECIFY R in Ohms" Z1
270 A1=1E+6*I/89/F/C1/E1
280 88=Z1*I/E1
290 Y=88: GOSU8 390: REM Y3=LN(Z1*I/E1)
300 X=0.6*Y3: GOSU8 520: REM X3=08 pwr 0.6
310 V2=E1*1.414*(1-83*X3)
320 V3=V2*(1-84*A1)
330 X=Y3/3: GOS 520: I1=I*85/X3: REM Ipk
340 X=Y3/6: GOS 520: I2=I*86/X3: REM Irms
350 GOTO 900
360 PRINT "F-W RECTIFIER WITH SERIES RESISTANCE"
370 81=180: 82=3.112: 83=1: 84=1: 85=1.46: 86=0.793: 89=6.283: 0=180
380 GOTO 180
385 REM SU8ROUTINE: Y3=LN(Y)
390 IF Y<0 PRINT "NEG OPO INVALIO": STOP
400 IF Y=0 Y3=-1E+37: RETURN
410 IF Y=1 Y3=0: RETURN
420 Y9=0: Y8=0
430 IF Y<1 Y=1/Y: Y8=1
440 IF Y>1 Y=Y/2: Y9=Y9+1: GOTO 440
450 Y=1-Y: Y1=16: Y2=2: Y3=Y: Y4=Y
460 Y4=Y4*Y: Y3=Y3+Y4/Y2
470 Y2=Y2+1: Y1=Y1-1: IF Y1>0 GOTO 460
480 IF Y2<0 Y3=Y3/2: Y2=Y2+1: GOTO 480
490 Y5=Y9*0.693148
500 Y3=Y5-Y3: IF Y8=1 Y3=-Y3
510 RETURN
515 REM SU8ROUTINE: X3=EXP(X)
520 X8=0: X9=0: IF X=0 X3=1: RETURN
530 IF X<0 X=-X: X8=1
540 IF X>1 X=X/2: X9=X9+1: GOTO 540
550 IF X=1 X3=2.718282: GOTO 620
560 REM X IS NOW 8ETWEEN 0 & 1
570 X3=X+1: X4=X: X1=8: X2=2
580 X4=X4*X/X2: X3=X3+X4
590 X2=X2+1: X1=X1-1: IF X1>0 GOTO 580
600 IF X9=0 GOTO 620
610 X3=X3*X3: X9=X9-1: IF X9>0 GOTO 610
620 IF X8=1 X3=1/X3
630 RETURN
635 REM SU8ROUTINE: Z3=ARCCOS(Z)
640 Z2=1: Z4=4
650 Z1=COSR(Z2)-Z
660 Z3=Z2+Z1/SINR(Z2)
670 Z4=Z4-1: Z2=Z3
680 IF Z4>0 GOTO 650
690 RETURN
700 PRINT "SPECIFY H-W OR F-W": A$=INPUT$
705 IF MIO$(A$,1,1)="H" GOTO 720
710 IF MIO$(A$,1,1)="F" GOTO 860
715 GOTO 700
```

```
 720 PRINT "H-W RECTIFIER WITH INDUCTIVE SERIES IMPEDANCE"
 725 B1=45.7: B4=0.926: B5=477: B6=1.04: B7=2.6: B8=1.415: B9=6.283: D=200
 730 B=B1/N/N/N/N: Y=B: GOSUB 390: REM Y3=LN(B)
 735 X=0.486*Y3: GOSUB 520: REM X3=Bpwr0.486
 740 E2=0.7777*(V1+2)/(1-R/200)/(1-B4*X3)
 745 L2=1000*B*E2/B9/F/I
 750 X=B6*Y3: GOSUB 520: REM X3=BpwrB6
 755 C2=B5*X3*1E+9/(B9*B9*F*F*L2*R)
 760 PRINT: PRINT "Emin =",E2: PRINT "Lmin =",L2: PRINT "Cmin =",C2
 770 PRINT: INPUT "SPECIFY E" E1, "INDUCTANCE in milliH" LI,
     "CAPACITANCE in microF" C1
 780 Y=B9*F*L1*I/(E1*1000): A=Y: GOSUB 410: REM Y3=LN(A)
 790 X=Y3*0.486: GOSUB 520: REM X3=Apwr0.486
 800 V2=1.414*E1*(1-B4*X3)
 810 X=Y3*B6: GOSUB 520: REM X3=A pwr B6
 815 X=X3*B5*1E+9/(B9*B9*F*F*L1*C1)
 820 V3=V2*(1-X/200)
 830 X=Y3/4: GOSUB 520: I1=I*B7/X3: REM Ipk
 840 X=Y3/8: GOSUB 520: I2=I*B8/X3: REM Irms
 850 GOTO 900
 860 PRINT "F-W RECTIFIER WITH INDUCTIVE SERIES IMPEDANCE"
 870 B1=5.772: B4=0.664: B5=106: B6=0.9: B7=1.55: B8=0.773: B9=6.283: D=200
 880 GOTO 730
 900 PRINT: PRINT "Vout =",V2, "Vmin =",V3, "Vno-load =",EI*1.414
 910 PRINT: PRINT "Diode Ipk =",I1
 915 PRINT: PRINT "Diode Irms =",I2
 920 PRINT: PRINT "r max =", D*(1-V3/V2)
 930 STOP
1200 PRINT "RECTIFIER LOADED BY RESISTOR R1"
1220 INPUT "SPECIFY ON-LOAD OUTPUT VOLTAGE" V1
1230 INPUT "SPECIFY MIN LOAD RESISTANCE IN OHMS" R1
1240 PRINT "MAX LOAD CURRENT I IN milliA =", 1000*V1/RI: PRINT
1250 INPUT "SPECIFY PK-PK RIPPLE VOLTAGE AS PERCENTAGE OF MEAN OUTPUT VOLTAGE.
     ANALYSIS VALID FOR RIPPLE NOT GREATER THAN 10%" R: PRINT
1260 PRINT "SPECIFY H-W OR F-W": A$=INPUT$
1270 IF MID$(A$,1,1)="F" GOTO 1500
1280 PRINT V1,"-VOLT H-W RECTIFIER WITH LOAD RESISTOR",R1,"OHMS": PRINT
1290 INPUT "DESIGN IMPLIES N=Ipk/I NOT LESS THAN 7.5 ENTER N" N: PRINT
1295 IF N<7.5 GOTO 1290: REM DO NOT ACCEPT LESS THAN 7.5 FOR N
1300 B1=24.67: B2=11.49: B3=10: B4=0.3: B5=2.6B1: B6=0.B8
1310 Y=N: GOSUB 390: X=Y3*3.25
1320 GOSUB 520: R2=R1*B1/X3
1330 Z=1-B2/N/N
1340 E1=V1/1.414/Z
1350 GOSUB 640: REM Z3=ARCOS(Z)
1360 C=1E+7*B3*(1-B4*Z3)/F/R/R1
1370 PRINT "Emin =",E1: INPUT "DESIGN E=" E1: PRINT
1380 PRINT "Cmin in microF =",C: INPUT "DESIGN C =" C: PRINT
1390 PRINT "Rmin on ohms =",R2: INPUT "DESIGN R =" R2: PRINT
1400 Y=R1/R2: GOSUB 390: X=0.3077*Y3
1410 GOSUB 520: N=X3*B5: Z=1-B2/N/N
1420 V2=E1*1.414*Z
1430 PRINT "MINIMUM OUTPUT VOLTAGE =",V2: PRINT
1440 PRINT "UNLOADED OUTPUT VOLTAGE =",V2/Z: PRINT
1450 PRINT "Diode Ipk in milliA =", I000*N*V2/R1: PRINT
```

```
1460 PRINT "Diode Irms in milliA =", 1000*V2*B6*SQR(N)/R1: PRINT
1470 GOSUB 640
1480 PRINT "PERCENT RIPPLE =", 1E+7*B3*(1-B4*Z3)/F/R1/C
1490 STOP
1500 PRINT V1,"-VOLT F-W RECTIFIER WITH LOAD RESISTOR", R1,"OHMS": PRINT
1510 INPUT "DESIGN IMPLIES N=Ipk/I NOT LESS THAN 3.75. ENTER N" N: PRINT
1515 IF N<3.75 GOTO 1510: REM DO NOT ACCEPT N VALUES LESS THAN 3.75
1520 B1=5.131: B2=2.856: B3=5: B4=0.54: B5=1.654: B6=0.623
1530 GOTO 1310
```

---------- THE END ----------

FLEXIBILITY & CREATIVITY TEST

This test does not measure your intellegence, your fluency with words and certainly not your mathematic ability. It will, however, give you some idea of your mental flexibility and creativity.
In the past three years since the test was developed, few people could solve more than half the 24 questions at the first attempt. Many, however, reported getting answers long after the test had been put aside, particularly at unexpected times, when their minds were relaxed.

TAKE THIS AS A PERSONAL CHALLENGE.

EXAMPLE:- 16 O In a P ANSWER:- 16 Ounces in a Pound.

1. 26 L of the A
2. 7 W of the AW
3. 1001 A N
4. 12 S of the Z
5. 54 C in a D (with the J)
6. 9 P in the SS
7. BB P K
8. 13 S on the AF
9. 32 D F at which W F
10. 18 H on a G C
11. 90 D in a R A
12. 200 P for P G in M
13. B S on a SS
14. 3 B M (S H T R)
15. 4 Q in a C
16. 24 H in a D
17. 1 W on a U
18. 5 D in a Z C
19. 57 H V
20. 11 P in a F T
21. 1000 W that a P is W
22. 29 D in F in a L Y
23. 64 S on a C
24. 40 D and N of the G F

HURKLE
FOR XTAL BASIC
BY NEIL SAUNDERS

(ED Neil is aged 12 and is welcome to our pages.)

```
 10 CLS
 20 PRINT TAB(33);"HURKLE"
 30 PRINTTAB(33);"======"
 40 PRINT
 50 PRINT TAB(20);"BY NEIL SAUNDERS 17/12/85 AGE 12"
 60 PRINT
 65 G=9
 70 PRINT@ 10,5, "A HURKLE IS HIDING ON A";G;"BY";G;"GRID. HOMEBASE"
 80 PRINT@ 10,6,"ON THE GRID IS POINT 0,0 AND ANY GRIDPOINT IS A"
 90 PRINT@ 10,7,"PAIR OF WHOLE NUMBERS SEPARATED BY A COMMA. TRY TO"
100 PRINT@ 10,8,"GUESS THE HURKLE'S GRIDPOINT. YOU GET 0-9 TRIES."
110 PRINT@ 10,9,"AFTER EACH TRY, I WILL TELL YOU THE APPOXIMATE"
120 PRINT@ 12,10,"DIRECTION TO GO LOOK FOR THE HURKLE"
130 QQ=&F320
140 FOR T=1 TO 10: QQ=QQ+&40: FOR I=I TO 10
150 POKE QQ+I,&1A: NEXT I,T
160 PRINT
170 PRINT@ 1,14,"HOW MANY GUESSES! (1-9) ";: N$=INCH$
171 N=VAL(N$)
180 A=RND(10)
190 B=RND(10)
200 PRINT@ 1,14,"                                    "
210 PRINT@ 28,16,"WEST": PRINT@ 44,16,"EAST": PRINT@ 35,12,"NORTH":
    PRINT@ 35,23,"SOUTH";
220 FOR K=1 TO N
230 PRINT@ 12,12,"          "
240 PRINT@ 5,12,"GUESS £";K;
250 INPUT Y,X
260 IF ABS(X-A)+ABS(Y-B)=0 THEN 390
270 REM PRINT INFO
280 GOSUB 430
290 PRINT
300 NEXT K
310 PRINT
320 CLS: PRINT@ 18,8,"SORRY, THAT'S";N;"GUESSES."
330 PRINT@ 18,10,"THE HURKLE WAS AT ";A;",";B
340 PRINT
350 PRINT@ 10,14,"DO YOU WANT TO PLAY AGAIN. HURKLE IS HIDING.";:
    Q$=INCH$: IF Q$="Y"THEN 10
360 CLS: PRINT@ 24,11,"";: END
370 PRINT
380 GOTO 180
390 REM
400 PRINT
410 CLS:  PRINT@ 18,8;"YOU FOUND HIM IN";K;"GUESSES!"
420 GOTO 340
430 PRINT@ 5,13,"              ": PRINT@ 5,13,"GO ";
440 IF Y=B THEN 490
450 IF Y<B THEN 480
```

```
460 PRINT"SOUTH";
470 GOTO 490
480 PRINT"NORTH";
490 IF X=A THEN 540
500 IF X<A THEN 530
510 PRINT"WEST";
520 GOTO 540
530 PRINT"EAST";
540 PRINT
550 RETURN
560 REM ********************
570 REM * BY NEIL SAUNDERS *
580 REM *     17/12/85     *
590 REM *      AGE 12      *
600 REM ********************
610 END
```

MORSE CODE READER
FOR INTERAK WITH A VDU-2K
BY MIKE WHARTON

From:- Mike Wharton, Swevenings, 8 Dvitts Close, Winslow, Bucks, MK18 3QQ.
Phone 029-671-4367

Dear Bob,

Here, at last, is the Morse Reader program for the Interak which I mentioned to you some time ago.

I hope its not too long to publish as source code, since this will make for easier modification by anyone brave enough to try running it.

The hardware requirements are "simply" a one-bit input port connected to D0, with the morse tone being used to represent a '1' and absence of tone for a '0' (Ed called the NRZI method I belive). Some extra bit of circuitry will be needed to do this if taking it from the audio output on the receiver, but I m sure most users could work this out for themselves.

The form as written is intended for operation on a 64 column screen, ie VDU-2K; I must confirm that this is a great improvement over the 32 column screen, I cannot imagine why I never had the courage to do the mod. ages ago! Incidentally, I reckon I can cut down the modding time to about 45 minutes by bending the various I.C. pins out at right angles and making all the extra connections by wire-wrapping: no cut tracks or de-soldering at all.

In course of preparation is a RTTY reader, but I have not yet had the chance to test it out properly; in either case, if any reader is sufficiently interested, I will gladly supply tape copies of both source and/or object code at 300/1200/2400 baud.

Finally, may I ask if any other user of Interak has any experience of using it to decode/display WEFAX transmissions from the American NOAA or TIROS type satellites, as this is the subject of my next project .

Many regards and thanks for an interesting news-letter, keep up the good work!

Mike Wharton

(Ed Thank you Mike for an excellent contribution. I would very much like to watch my machine typing out RTTY transmissions in English as they came in from the radio - facinating stuff, can we have more please.)

```
                ;Morse Code Reader
                ;For INTERAK with VDU-2K
                  ORG  5000H      ;Put it elsewhere if you like
                         LOAD 5000H     ; and remember to change this line too
E6E2              K8Q      EQU 0E6E2H      ;ZYMON keyboard routine
F000              SCRN     EQU 0F000H      ;VDU-2K
0018              LINES    EQU 24
0040              COLS     EQU 64
0005              PORT     EQU 5
5000  CDFC50      ENTER:   CALL CVDU       ;Clear screen
5003  CD2851               CALL SIGNON
5006  C00150               CALL PRINT
5009  01010C      START:   LD BC,0C01H
500C  210000               LD HL,0
500F  C09B50               CALL GETCHR
5012  2034                 JR NZ,TRY4
```

```
5014  24        NEXT:   INC H
5015  7C                LD A,H
5016  CB3F              SRL A
5018  CB3F              SRL A
501A  CB3F              SRL A
501C  B8                CP 8
501D  3807              JR C,TRY1
501F  CD9B50    AGAIN:  CALL GETCHR
5022  38E5              JR C,START
5024  18F9              JR AGAIN

5026  CD9B50    TRY1:   CALL GETCHR
5029  38E9              JR C,NEXT
502B  2C        TRY2:   INC L
502C  78                LD A,B
502D  CB3F              SRL A
502F  BD                CP L
5030  380C              JR C,TRY3
5032  CD9850            CALL GETCHR
5035  30F4              JR NC,TRY2
5037  7C                LD A,H
5038  85                ADD A,L
5039  67                LD H,A
503A  2E00              LD L,0
503C  18D6              JR NEXT

503E  7C        TRY3:   LD A,H
503F  CB3F              SRL A
5041  B8                CP 8
5042  CB11              RL C
5044  2600              LD H,0
5046  1809              JR TRY5

5048  2C        TRY4:   INC L
5049  7D                LD A,L
504A  CB3F              SRL A
504C  CB3F              SRL A
504E  B8                CP B
504F  301B              JR NC,TRY7
5051  CD9B50    TRY5:   CALL GETCHR
5054  30F2              JR NC,TRY4
5056  24        TRY6:   INC H
5057  78                LD A,B
5058  CB3F              SRL A
505A  BC                CP H
505B  3829              JR C,TRY11
505D  CD9B50            CALL GETCHR
5060  38F4              JR C,TRY6
5062  7D                LD A,L
5063  BC                ADC A,H
5064  6F                LD L,A
5065  2600              LD H,0
5067  18DF              JR TRY4
5069  79        TRY7:   LD A,C
506A  FE01              CP 1
```

```
506C  2806            JR Z,TRYB
506E  CDB150          CALL DUTCHR
5071  CDB150          CALL DUTCHR
5074  2E00      TRY8: LD L,00
5076  2600      TRY9: LD H,00
5078  CD9B50    TRY10:CALL GETCHR
5078  30F9            JR NC,TRY9
507D  24              INC H
507E  7B              LD A,B
507F  CB3F            SRL A
50B1  BC              CP H
50B2  30F4            JR NC,TRY10
50B4  1BA0            JR TRY1

50B6  78        TRY11:LD A,8
50B7  CB3F            SRL A
50B9  80              ADD A,B
50BA  BD              CP L
50BB  3B07            JR C,TRY12
50BD  7B              LD A,B
50BE  85              ADD A,L
50BF  CB3F            SRL A
5091  47              LD B,A
5092  1B03            JR TRY13

5094  CD8150    TRY12:CALL DUTCHR
5097  2E00      TRY13:LD L,0
5099  1B88            JR TRY1

5098  C5        GETCHR:PUSH BC
509C  0605            LD 8,5
509E  0E00            LD C,0
50A0  DB05      INPUT:IN A,(PDRT)
50A2  E601            AND 1
50A4  B1              ADD A,C
50A5  4F              LD C,A
50A6  CDF050          CALL DELAY
50A9  0F              RRCA
50AA  10F4            DJNZ INPUT
50AC  79              LD A,C
50AD  FE03            CP 3
50AF  C1              PDP BC
50B0  C9              RET

50B1  C5        DUTCHR:PUSH BC
50B2  E5              PUSH HL
50B3  79              LD A,C
5084  013200          LD BC,32H
50B7  213551          LD HL,TABBGN
50BA  EDB1            CPIR
50BC  013100          LD BC,31H
50BF  09              ADD HL,BC
50C0  7E              LD A,(HL)
50C1  CDD150          CALL PRINT
50C4  CDE2E6          CALL KBD
```

```
50C7  FE5A              CP "Z"          ;Hit Z to escape to ZYMON
50C9  CA6600            JP Z,66H
50CC  E1                POP HL
50CD  C1                PDP BC
50CE  0E01              LD C,1
50D0  C9                RET

50D1  E5        PRINT:  PUSH HL
50D2  D5                PUSH DE
50D3  213451            LD HL,COUNT
50D6  16F5              LD D,0F5H
50DB  5E                LD E,(HL)
50D9  12                LD (DE),A
50DA  3E5F              LD A,5FH
50DC  13                INC DE
50DD  12                LD (DE),A
50DE  1B                DEC DE
50DF  7B                LD A,E
50E0  FEBF              CP 0BFH
50E2  2009              JR NZ,EXIT
50E4  CD0A51            CALL SCRDLL
50E7  3EB0              LD A,B0H
50E9  77        STORE:  LD (HL),A
50EA  D1                POP DE
50EB  E1                PDP HL
50EC  C9                RET

50ED  3C        EXIT:   INC A
50EE  1BF9              JR STORE

50F0  D5        DELAY:   PUSH DE
50F1  119000             LD DE,90H
50F4  15        REP1:    DEC D
50F5  20FD               JR NZ,REP1
50F7  1D        REP2:    DEC E
50FB  20FD               JR NZ,REP2
50FA  D1                 PDP DE
50FB  C9                 RET

50FC  2100F0    CVDU:    LD HL,SCRN
50FF  1101F0             LD DE,SCRN+1
5102  015B00             LD BC,LINES+1*CDLS
5105  3620               LD (HL),20H
5107  EDB0               LDIR
5109  C9                 RET

510A  E5        SCRDLL:  PUSH HL
510B  D5                 PUSH DE
510C  C5                 PUSH BC
510D  F5                 PUSH AF
510E  1100F0             LD DE,SCRN
5111  2140F0             LD HL,SCRN+CDLS
5114  01DBFF             LD BC,LINES-1*COLS
5117  EDB0               LDIR
5119  21DBEF             LD HL,LINES-1*COLS+SCRN
```

```
511C  3620      CLR:    LD (HL),20H
511E  7D                LD A,L
511F  E63F              AND 3FH
5121  20F9              JR NZ,CLR
5123  F1                POP AF
5124  C1                POP BC
5125  D1                POP DE
5126  E1                POP HL
5127  C9                RET

512B  219D51    SIGNDN: LD HL,MSG
512B  115BEF            LD DE,LINES-3*CDLS+SCRN
512E  011900            LD BC,25
5131  EDB0              LDIR
5133  C9                RET

5134  B0        CDUNT:  DB B0H
5135  01061715  TABBGN: DB 1,6,17H,15H
5139  0B031D09          DB 0BH,3,1DH,9
513D  1F071B0A          DB 1FH,7,1BH,0AH
5141  1B04050B          DB 1BH,4,5,B
5145  19120D0F          DB 19H,12H,0DH,0FH
5149  020E1E0C          DB 2,0EH,1EH,0CH
514D  16141330          DB 16H,14H,13H,30H
5151  3B3C3E3F          DB 3BH,3CH,3EH,3FH
5155  2F272321          DB 2FH,27H,23H,21H
5159  202E6A2D          DB 20H,2EH,6AH,2DH
515D  4C35BA7A          DB 4CH,35H,0BAH,7AH
5161  73475552          DB 73H,47H,55H,52H
5165  3700              DB 37H,0

5167  20414243  TABLE:  DB 20H,41H,42H,43H
516B  44454647          DB 44H,45H,46H,47H
516F  4B494A4B          DB 4BH,49H,4AH,4BH
5173  4C4D4E4F          DB 4CH,4DH,4EH,4FH
5177  50515253          DB 50H,51H,52H,53H
517B  54555657          DB 54H,55H,56H,57H
517F  5B595A31          DB 5BH,59H,5AH,31H
51B3  32333435          DB 32H,33H,34H,35H
51B7  36373B39          DB 36H,37H,3BH,39H
51BB  303D2E2F          DB 30H,3DH,2EH,2FH
51BF  2C232D3C          DB 2CH,23H,2DH,3CH
5193  3F3A3B29          DB 3FH,3AH,3BH,29H
5197  222A              DB 22H,2AH
5199  00000000          DB 0,0,0,0

519D  494E5445  MSG:    DB "INTERAK MORSE READER....."
      52414B20
      4D4F5253
      45205245
      41444552
      2E2E2E2E
      2E

                        END
```

DISSASSEMBLER MOD
BY STEPHEN HARGREAVES

From:- Stephen Hargreaves, 12 Portland Place, Altricham, Cheshire, WA14 2PA.
Dear Ed,

I enclose an assembly listing of some modifications to the "HC DISASS" disassembler, which you may like to publish. These mods allow output to be sent to the printer: The printer is turned on and off by CTRL P.

Also, the destinations of relative branches are printed instead of the offset, and ASCII characters of the data being disassembled are sent to the printer.

Secondly, is there a bug in ASM64? When using the assembler, the source text occasionally becomes corrupted so that it cannot be assembled.

When the offending area of text is listed on the screen, it looks as though a cursor control character has been inserted in the text.

This fault can be rectified by re-typing some of the lines of text, but I never noticed this problem with ASM32, and it seems to occur randomly.

Yours Faithfully
Stephen Hargreaves.

(ED:- Thanks for the mods. ASM64 is an updated (for screen size only) version of ASM32 so the code is very much the same. Software bugs cannot occur at random. If a bug exists in a piece of code then every time that sequence is executed the bug will show up. It cannot fix itself sometimes, it will allways produce the same fail for the same pass. If a bug exists, the way to trace it is to determine precisly which sequence of events preceed the bug symptoms appearing. More probally the program that you are constructing with ASM64 is altering some part of ASM64s source area. Get the bug to show. Then reload and try to pin down exactly at which point in the sequence the bug appears. With any luck it can be pinned down to a particular command which when executed produces the bug. Finally are you sure that you are not LOADing to part of your own source file, it runs upwards from 3200H and it is possible to corrupt it by overwriting it with object code. This modification that you have enclosed runs very close to its own source file! (ie 403CH). Check before assembly with the M command.)

```
                ;DISSASSEMBLER MOD
                ;=================

                ;Make dissassembler output to printer,
                ;calculate destinations of relative branches,
                ;and print ASCII characters of bytes.

0602            SCROLL: EQU 602H
06E2            KEYBD:  EQU 6E2H
00E0            LINEMS: EQU 0E0H

                ;First patch over original dissassembler subroutine calls

403C                    ORG  403CH
                        LOAD 403CH
403C  CDA14A            CALL LNPT

4013                    ORG 4013H
                        LOAD 4013H
```

```
4013  CD0B4B            CALL INCH
4016  00                NOP
4017  00                NOP

40B0                    ORG 40B0H
                        LOAD 40B0H
40B0  CD0B4B            CALL INCH
40B3  00                NOP
40B4  00                NOP

42DB                    ORG 42DBH
                        LOAD 42DBH
42DB  CD1A4B            CALL DFST

419F                    ORG 419FH
                        LOAD 419FH
419F  CD1A4B            CALL DFBT

41B9                    ORG 41B9H
                        LOAD 41B9H
41B9  CD1A4B            CALL DFBT

                ;In place of original scroll call,
                ;copy line to line printer as well

4AA1                    ORG 4AA1H
                        LOAD 4AA1H
4AA1  F5        LNPT:   PUSH AF
4AA2  C5                PUSH BC
4AA3  D5                PUSH DE
4AA4  3AFD4A            LD A,(PFLAG)    ;PRDN or PRDF?
4AA7  FE00              CP 0
4AA9  2B4B              JR Z,LNPTZ
4AAB  01C0F2            LD BC,0F2C0H    ;Copy line to printer
4AAE  0A        LDDP1:  LD A,(BC)
4AAF  57                LD D,A
4AB0  CDFE4A            CALL PRNT
4AB3  03                INC BC
4AB4  79                LD A,C
4AB5  FEE0              CP LINEMS
4AB7  20F5              JR NZ,LDDP1
4AB9  01C6F2            LD BC,0F2C6H    ;Convert bytes to ASCII
4ABC  0A        LDDP2:  LD A,(BC)
4ABD  D630              SUB 30H
4ABF  FE0A              CP 0AH
4AC1  3B02              JR C,L2
4AC3  D607              SUB 7
4AC5  CB27      L2:     SLA A
4AC7  CB27              SLA A
4AC9  CB27              SLA A
4ACB  CB27              SLA A
4ACD  57                LD D,A
4ACE  03                INC BC
4ACF  0A                LD A,(BC)
4AD0  D630              SUB 30H
```

```
4AD2  FE0A              CP 0AH
4AD4  3B02              JR C,L3
4AD6  D607              SUB 7
4ADB  B2          L3:   ADD A,D
4AD9  FE20              CP 20H          ;all characters printed?
4ADB  3004              JR NC,L4
4ADD  3E23              LD A,23H
4ADF  1B06              JR L5
4AE1  FE7F        L4:   CP 07FH
4AE3  3B02              JR C,L5
4AE5  3E23              LD A,23H        ;If not a printable charater
4AE7  57          L5:   LD D,A          ;print £ instead
4AEB  CDFE4A            CALL PRNT
4AEB  03                INC BC
4AEC  0A                LD A,(BC)
4AED  D621              SUB 21H
4AEF  30CB              JR NC,LDDP2
4AF1  160D              LD D,0DH
4AF3  CDFE4A            CALL PRNT
4AF6  CD0206      LNPTZ:CALL SCRDLL
4AF9  D1                PDP DE
4AFA  C1                PDP BC
4AFB  F1                PDP AF
4AFC  C9                RET

4AFD  00          PFLAG:DB 0

4AFE  DB06        PRNT: IN A,(6)          ;Sends D to printer
4B00  CB7F              BIT 7,A
4B02  2BFA              JR Z,PRNT
4B04  7A                LD A,D
4B05  D307              OUT (7),A
4B07  C9                RET

4B0B  CDE206      INCH: CALL KEYBD        ;Get character from keyboard
4B0B  2BFB              JR Z,INCH
4B0D  FE10              CP 10H
4B0F  C0                RET NZ
4B10  3AFD4A            LD A,(PFLAG)
4B13  EEFF              XOR 0FFH
4B15  32FD4A            LD (PFLAG),A
4B1B  1BEE              JR INCH

4B1A  F5          DFST: PUSH AF           ;Convert relative branch offset
4B1B  C5                PUSH BC         ;to absolute destination address
4B1C  23                INC HL
4B1D  E5                PUSH HL
4B1E  7E                LD A,(HL)
4B1F  D5                PUSH DE
4B20  ED5BC30F          LD DE,(0FC3H)
4B24  CDDB4B            CALL 4BDBH
4B27  CDF94B            CALL 4BF9H
4B2A  ED53C30F          LD (0FC3H),DE
4B2E  D1                PDP DE
4B2F  E1                PDP HL
```

```
4B30  E5              PUSH HL
4B31  7E              LD A,(HL)
4B32  0600            LD B,0
4B34  CB7F            BIT 7,A
4B36  2B02            JR Z,OFST1
4B3B  06FF            LD B,0FFH
4B3A  4F       OFST1: LD C,A
4B3B  09              ADD HL,BC
4B3C  23              INC HL
4B3D  7C              LD A,H
4B3E  CDDB4B          CALL 4BDBH
4B41  CDF94B          CALL 4BF9H
4B44  7D              LD A,L
4B45  CDDB4B          CALL 4BDBH
4B4B  CDF94B          CALL 4BF9H
4B4B  E1              POP HL
4B4C  C1              POP BC
4B4D  F1              POP AF
4B4E  C9              RET

                      END
```

INTERAK BUGGY
BY STEVE PADLEY

From:- Steve Padley, 14 Wickham Rd, Fareham, Hants, PO16 7EU.
Dear Ed,

This is for anybody interested in robotics and computer controlled equipment.

This first article is about a module developed to be a stepper motor controller for a small buggy or 'turtle' but it also has the address decoding circuitry for expansion.

MOTOR CONTROLLER AND DECODING BOARD
£££££££££££££££££££££££££££££

The board circuit diagram is split into two sections :-

1. Power supply and decoding section.
2. Motor controller section.

Probally the best way to describe this board is to talk you through the circuit diagram shown on Fig 1.

Eventually the module will run via a serial interface but at the moment it is fed with parallel data via my progammable 8255 Parallel interface chip previously described in an earlier article.

This module is built on a piece of V-Q board which is 58 holes x 28 rows. Parallel data, 0v line and a 12v rail is fed to the module via a 16 way board mounted IDC connector. This is fed to 2 sets of 10 way minicon terminals so that all the power and data lines can be fed to other boards piggy backed to this one as the buggy expands.

POWER SUPPLY

The 12v rail is required to drive the stepper motors and the motor control chips. Initially this must be able to provide a current of 1A at the very least until the time comes for the buggy to expand. The 7805 regulator provides the stable 5v for the TTL on this and the future boards. Hence the necessity for a healthy 12v rail.

DATA LINES AND DECODING

The 8 data lines are divided into 2 nibbles, D0-D3 are used to send data to or receive data from onboard devices, this board requires data for motor speed and direction control. D4-D6 are fed to the inputs of a 74LS138 (3 line to 8 line decoder) this allows us to access up to 8 individual circuits, eg Motors, Sleeper, Line detectors, Impact sensors, Fork lift of Pen lift and maybe even a speech synthesiser.

Y0 of the 138 is fed to the enable pin of the 74LS367 chip D0-D3 are also fed to the driver inputs of this chip as shown, thus when Y0 is low, data on D0-D3 is allowed through to the motor circuits. Y0 will be low when D4, D5 & D6 are all low, D7 as yet is not used.

All the other outputs of the 138 are fed to the second 10 way minicon terminal to access future circuits.

STEPPER MOTOR CONTROLLER

There are two identical circuits for the left and right motors. For simplicity one circuit will be described. The circuit is based around the SAA 1027 stepper motor controller chip and the RS Size 1 motor.

The SAA1027 will provide the coil codes to produce rotation by a single clock input on pin 15, the faster the clock the greater the speed up to the motors skip speed.

As can be seen on the diagram D0 privides the clock for left motor, so this bit is merely toggled (see example programs). Direction of rotation is governed by the voltage on pin 3 of the chip (0v one way, 12v the other).

LEDs are provided as direction indicators or blinking eyes as the motor does not have to be in motion when the LEDs are lit.

The BC109's are used to interface TTL 5v logic to 12v signals required by the SAA 1027 chip.

The value of resistor between pin 4 of SAA 1027 and the 12v rail sets the coil currents to the motor (200mA with the resistor shown), the lowest value resistor can be 130R, this will allow a current of 400mA, but my 12v supply needs to be upgraded first.

6 way minicom terminals are used to connect the motors to outputs of the chip.

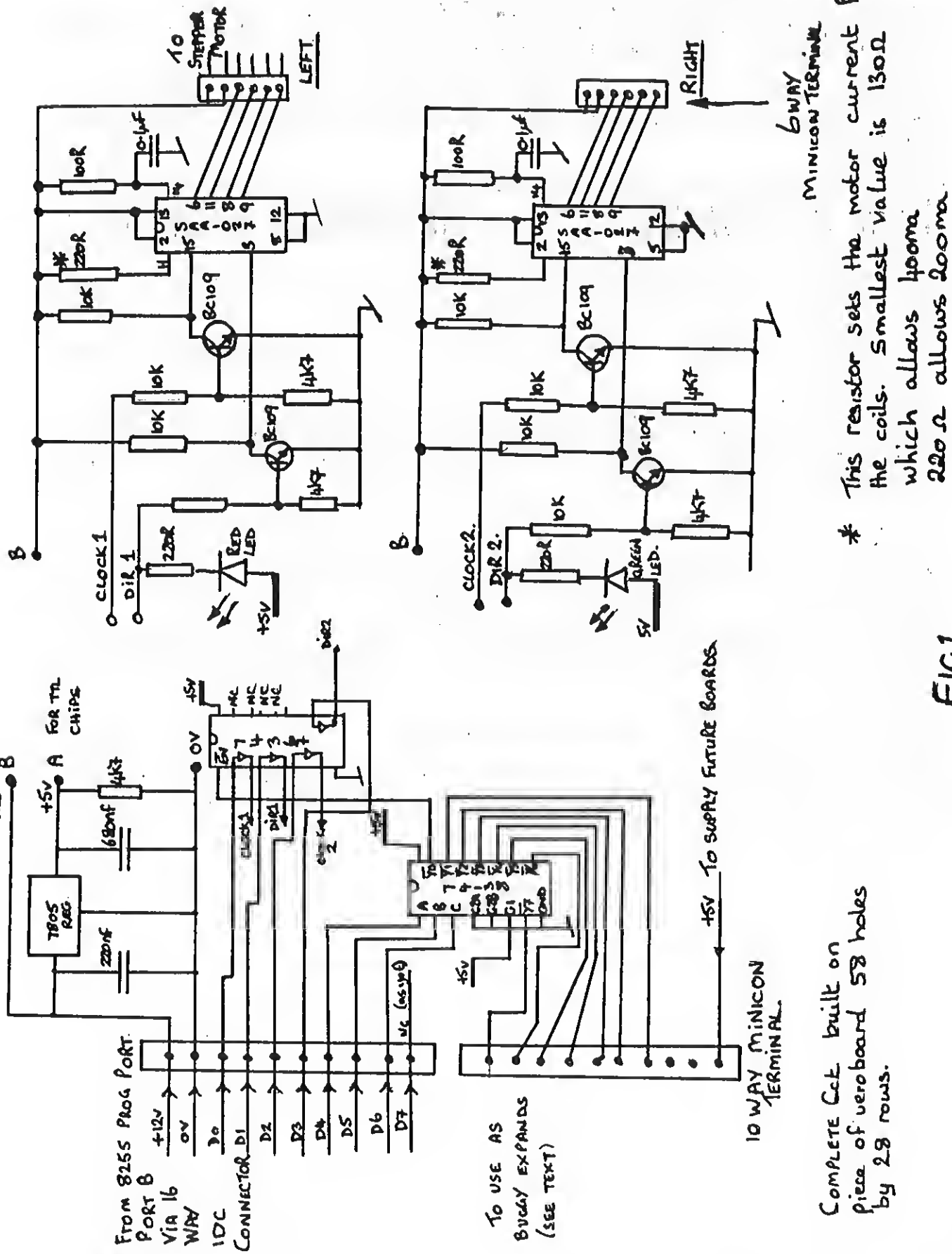
Care must be taken to ensure that the motor is correctly connected to the coil outputs from the chip otherwise rotation will not be achived.

Enclosed are a couple of test programs in Basic to check out the board and help you to understand how the thing works. A logic probe would be handy but a meter will do.

Next part will be a 'turtle' base and gearing if necessary.

Anybody interested in this as a project I have the data sheets for the SAA 1027 & RS Size 1 motors which I will be happy to duplicate for them at the cost of postage and duplication.

STEPPER MOTOR CONTROL FOR INTERAK BUGGY



BUGGY TEST PROGRAMS

```
10 REM MOTOR BOARD TEST1
20 REM -- TEST DIRECTION INDICATORS
30 OUT 15,128: REM--B255 PORT ACTIVATE (ALL PORT O/P'S)
40 OUT 13,0:   REM--PORT B BOTH LED'S SHOULD LIGHT
50 GOSUB 120
60 OUT 13,2:   REM--PORT B RIGHT LED ON
70 GOSUB 120
80 OUT 13,8:   REM--PORT B LEFT LED ON
90 GOSUB 120
100 OUT 13,10: REM--PORT B BOTH LED'S OFF
110 GOTO 200
120 FOR X=1 TO 1000: NEXT X
130 RETURN
200 REM MOTOR BOARD TEST 2
210 REM--LEFT MOTOR
215 A=0: B=1: S=0: REM--SET DATA FOR TOGGLING DD ANTICLOCKWISE
220 FOR TIME=1 TO 500: REM--TIME MOTOR TURNS FOR
230 OUT 13,A
240 FOR PAUSE=1 TO 5: NEXT PAUSE: REM--SPEED OF MOTOR
250 OUT 13,B
255 IF S=2 THEN 280 ELSE 260: REM--IS TEST COMPLETE
260 NEXT TIME
270 A=2: B=3: S=S+1: GOTO 220: REM--CHANGE DIRECTION CLOCKWISE
280 REM
300 REM MOTOR BOARD TEST 3
310 REM--RIGHT MOTOR
315 A=4: B=0: S=0
320 FOR TIME=1 TO 500
330 OUT 13,A
340 FOR PAUSE=1 TO 5: NEXT PAUSE
350 OUT 13,B
355 IF S=2 THEN 380 ELSE 360
360 NEXT TIME
370 A=12: B=8: S=S+1: GOTO 320
380 END
400 REM--ARMED WITH THIS INFORMATION SEE IF YOU CAN WRIGHT
410 REM--A TEST PROGRAM TO DRIVE BOTH MOTORS AT THE SAME TIME?
```

N.B. If driving both motors causes them not to rotate correctly you're 12v PSU is probally not capable of delivering enough current.

--------- THE END ----------

STARS
FOR XTAL BASIC
BY BRUCE JOYCE

```
10 CLS
20 GOSUB400
30 N$="     "
40 LETN=1: LETTT=0
50 GOSUB610
60 LETN=N+1: LETT=0: LETSCR=0: LETTIM=TIM1
70 LETA=10: LETB=10: LETF=0
80 PRINT@51,0;MUL$("~",10): PRINT@51,1;"STARS LEFT": PRINT@51,2;"    ";N-T
90 PRINT@51,3;MUL$("~",10): PRINT@51,4;)"   SCORE ": PRINT@51,5;">>       <<"
100 PRINT@51,6;MUL$("~",10): PRINT@51,8;"   TOTAL": PRINT@51,9;">>       <<"
110 PRINT@51@,10;MUL$("~",10): PRINT@51,11;"TOP SCORER":
    PRINT@51,12;">         <": PRINT@51,13;"   WITH": PRINT@51,14;">        <"
120 PRINT@51,15;MUL$("~",10): PRINT@51,16;" TIME LEFT":
    PRINT@51,17;">>       <<": PRINT@51,18;MUL$("~",10);
125 PRINT@51,19;"  LEVEL";: PRINT@51,20;">>       <<";: PRINT@51,21;
    MUL$("~",10);:
130 PRINT@56,5;200-SCR: PRINT@53,9;TT: PRINT@53,12;N$: PRINT@53,14;NT;:
    PRINT@55,20;N-2
140 LETMOVE=RND(4)
150 IFMOVE=0THENLETA=A+1
160 IFMOVE=1THENLETA=A-1
170 IFMOVE=2THENLETB=B-1
180 IFMOVE=3THENrLETB=B+1
190 IFA<1THENLETA=1
200 IFB<1THENLETB=1
210 IFA>49THENLETA=49
220 IFB>21THENLETB=21
230 IFF=30THENGOTO310
240 IFPEEK(&F000+A+(B*64))<>32THENA=C: B=D: F=F+I: GOTO140
245 TIM=TIM-1: FMT4,0: PRINT@53,17;TIM;: FMT0,0: IFTIM=0THENGOTO310
250 F=0
260 PRINT@A,B;"*"
270 PRINT@C,D;" "
280 IFKBD=32THENPRINT@C,D;"": LETSCR=SCR+I: PRINT@55,5;200-SCR
290 LETC=A: LETD=B
300 GOTO150
310 T=T+1: TT=TT+(200-SCR): PRINT@53,9;TT
315 IFTIM=0THENGOTO340
320 IFT<=NTHENGOTO70
330 IFN<=TARTHENGOTO50
340 CLS: IFNT>=TTTHENPRINT@10,10;"You have scored ";TT;".":
    FORX=1TO2000: NEXTX: CLS: GOTO40
345 IFTIM=0THENPRINT@10,10;"SORRY!TIME'S UP!YOU SCORED";TT
350 IFNT<TTTHENPRINT@10,12;"THAT'S GREAT!You scored ";TT;"."
360 NT=TT
370 INPUT"Type in your name.(No more than 6 letters please)";N$
380 CLS: GOTO40
390 END
400 PRINT@20,5;"
410 PRINT@20,6;"
420 PRINT@20,7;"
430 PRINT@20,8;"
```

```
440 PRINT@20,9;"
450 PRINT: PRINT: PRINT: PRINT: PRINTTAB(15)"Do you need instructions(Y/N)";:
    Y$=INCH$
460 IFY$="N"THENGOTO5B0
470 PRINT@0,12;"The object of the game is to catch stars.
4B0 PRINT"These stars race around the screen and can only be caught by
    trapping them.                                          "
490 PRINT"This is done by pressing the space bar and thus causing the
    star to leave a block behind it.
500 PRINT"When you have trapped the stars on one level then you advance
    to the next.This has one more star than before.
510 PRINT"But there is a limit to how many blocks you can force the
    star to leave on each level.
520 PRINT"Your score is the no.of blocks left":
    PRINT"Press any key to continue";: Z=INCH
530 FORX=12TO23: PRINT@0,X;SPC(64): NEXTX
540 PRINT@0,12;"The star must be trapped like this:-"
550 PRINT
    "                                                                    *B  "
560 PRINT"                "
570 PRINT"You have also to worry about the amount of time that is left.
    This is restarted at each level.
5B0 PRINT: PRINT"Please select time limit from 1000 to 5000.":
    TIM1=VAL(INCH$)*1000: IFTIM1<1000ORTIM1>5000THENGOTO5B0
590 PRINT: PRINT"please select the no.of levels that you want to play to,
    between 1 and 9.": TAR=VAL(INCH$)+1
600 CLS: RETURN
610 PRINT@0,0;MUL$("",50)
620 FORX=1TO22
630 PRINT"";TAB(50)""
640 NEXTX
650 PRINT@0,22;MUL$("",50)
660 RETURN
670 R=INCH: R=R-48: PRINTR
```

LETTERS

Please write with comments, ideas, complaints and suggestions. Name and address must be enclosed, but can be with-held. Responsibility for views and comments expressed cannot be held by the editor as members letters are published with the minimum changes (deleted bad language etc). (Note; I type what I see, if you forget a word then it will be missing in the newsletter, also if you spell a word wrong then it's quite likely that I will punch it in wrong.)

FROM :- SIMON WALLER,FLAT 2,MARITIME COURT,58 FOUNDATION ST, IPSWITCH,IP4 1BN
Dear Ed,

Now that there is a growing body of Interak users who have, or shortly will have, floppy disks and CP/M running on their systems, would it be possible to extend the software library to include disks as well as tapes. It is very irksome to have spent many hours of labour and pounds of money getting the disks going and then to find that all the available programs must be loaded from tape !!.

I would be willing to add to the library, on disk, programs to transfer data between tape and disk, Sargon (a chess program), and an adventure game.

Once an Interaker has CP/M running with a version of the BIOS, he (or she) might find that the CONOUT routine published in the newsletter a few issues ago has room for improvement, specifically to make it more intelligent. I have written my CONOUT routine to look like a VT52 terminal, which enables Wordstar to send commands like Insert Line and Clear To End of Line. I have listed the more interesting commands below :-

VDU Control Codes :-

08H	H	BACKSPACE
09H	I	TAB
0AH	J	LINE FEED
0DH	M	CARRIAGE RETURN

ESCape codes :- (i.e 1BH followed by the character)

2BH	(	CURSOR OFF
29H	)	CURSOR ON
2BH	+	CLEAR ALL TO SPACES
3FH	?	READ CURSOR POSITION (Return D=ROW, E=COLUMN)
41H	A	CURSOR UP
42H	B	CURSOR DOWN
43H	C	CURSOR RIGHT
44H	D	CURSOR LEFT
45H	E	LINE INSERT
48H	H	CURSOR HOME
4AH	J	CLEAR TO END OF SCREEN
4BH	K	CLEAR TO END OF LINE
50H	P	PRINT PAGE
51H	Q	INSERT CHARACTER
52H	R	DELETE LINE
54H	T	LINE ERASE TO SPACES
55H	U	PAGE ERASE TO SPACES
57H	W	CHARACTER DELETE
59H	Y	LOAD CURSOR POSITION (Followed by ROW+20H, COLUMN+20H)
5AH	Z	READ CHARACTER (Return in C)
65H	e	BACK TAB
69H	i	TAB
6AH	j	START REVERSE VIDEO
6BH	k	END REVERSE VIDEO

Read cursor position and Read character may only be used by calling DCONIO rather than the usual CONOUT. Obviously not all of these codes need be implemented but perhaps these could be used as a starting point from which to

develop a standard. The adventure game I mentioned earlier uses some of these codes to move around the screen and update the displayed information.

Yours sincerely
Simon Waller.

(ED:- Yes Simom I agree with you and I think a disk library will eventually be started. We need a librarian of course, perhaps you could consider it? contact me again with your requirements if you feel able to help. In the meantime I suggest you join the CP UK USERS GROUP. This group maintains a vast, and cheap, disk based software library. Here is the address :- CP/M USERS GROUP (UK), 72 MILL ROAD, HAWLEY, DARTFORD, KENT, DA2 72Z. Or phone then on 0322-22669. On the subject of terminal programs. Remember that all screens are memory mapped. Serial terminals are memory mapped devices with the subject memory seperated from the computer by, another computer, within the terminal, that simulates the commands that you list. Mind you, you are absolutly correct in saying that we need some real standards to build to. But who chooses them, and when are they to be chosen. Remember Interak is a homebrew computer, you build your way, else you go and buy any twin disk box and be standardised but locked in (which is worse?). Finally I wonder if you would like to submit your improved CONOUT routine for a future issue.)

****

FROM:- ERIC FURNESS, 23 ARMTHORPE RD, SHEFFIELD, S11 7FA.
Dear Editor,

In your editorial in Issue 6 you entered a plea for a new VDU board to replace the VDU-K. Could I take this a bit further and suggest that a standard is required for the whole system.

We have at the moment two screen formatts, two Basics and no standard DOS apart from the CP/M option. In addition I have no doubt that many members will, like myself, have added other cards to replace those which are not yet available from Greenbank. How about the User Group producing a specification for an 'ideal' system to work towards over the next few years?

My own inclination is to standardise on XTAL Basic; Ive used the 2.2 Version for several years and find it excellent. It does however, present further possibilities for non-standard additions to the reserved word list and action to prevent this could assure inter-changeability of programs in the future.

Could you tell me where you got your copy of the Data sheet for the WD2797 FDC chip? I,ve been trying, without any success, to get hold of a copy for some weeks now!.

Yours Sincerely
Eric Furness

(ED:- The standard DOS is CP/M, unless you can afford IBM clones etc. The standard for the hardware, in my opinion and as long as the Z80 chip lasts, is quite simple, 4MHZ, 64K, 2 x 3.5" drives running CP/M 2.2 to an 80 col. terminal card, with a LQ Epsom printer. As to a standard Basic. Yes I agree, but Basic is supposed to be a standard in itself. The pity is that all these clever bits have been added to enhance the lock-in factor and reduce the language to a computer typed system. A skilled programmer needs only about 20 standard keywords coupled with clever coding. As soon as you drift into second level keywords you destroy compatability and standardisation becomes a

joke. Perhaps the best way forward will be for us to move to the 'C' language as it has not yet been destroyed by over clever implimentations. A full data sheet for the 2797 chip has been published in IUGN 11).

****

FROM:- ERROL PAGE, THE BURROWS, SUMMERLUG, MORTIMER, READING, RG7 2JR.
Dear Ed,

Could I use the newsletter for an ad. please? Has anyone any assembled, or even part assembled, Interak cards that they may wish to sell? I am interested also in a keyboard if someone in the user group has one surplus to requirements. My progress with building an Interak has been extremely slow and perhaps this might speed things up a bit.

Talking about surplus to requirements I have a number of used Athena double-sided, double-density, 8" floppy disks which I would like to sell. They are formatted single-density CP/M and verified but not sysgened. Some are in sets of 10 (A to J).

I am asking £8.00 for 10 or 90p each. If anyone is interested would they phone 0734-332642 in the evening, please.
Thank you and other contributors for producing in 'Interaktion' a very interesting newsletter.

Yours sincerely,
Errol Page.

****

FROM:- DAVE GORDON, 229 STONELOW RD, DRONFIELD, SHEFFIELD, S18 6ER.
Dear Bob,

Further to previous letters about modems could I use this one to sing praises for Tom Evans, your new Treasurer (Ed and membership sec). As you know we two have stayed in touch with each other quite a bit since our first Interak to Interak modem connection. Tom now runs a Bulletin Board which welcomes Interakers as well as other constructors. I know that Tom has personally put an awful lot of effort into this Bulletin Board. He has helped me enormously and I hope to have a Bulletin Board up soon. On behalf of myself and all users of Toms Bulletin Board, thank you Tom.

Do the User Group intend to utilise Tom's Board for official information exchange etc? Will you be publishing his details? phone no etc. If so then it might be a good idea to print his menu's in the newsletter cos this will speed up use of the Bulletin Board and thus save the caller money.

I have lots of idea's which to me seem quite exciting but am unsure of the feeling of other Interakers. So far only a handfull have expressed an interest in getting their Interak on-line. Is this a true picture.

How does the number of Interaks sold compare with the number in the User Group? If there was a large difference is there any point trying a renewed effort to gain more U.G members? Thank you for your articles in the User Group newsletter, I have found them all to be of enormous value.

Yours Sincerely
Dave Gordon

(ED:- TOM's BULLETIN BOARD is excellent and I belive the membership is

growing quite steadily, you must remember that not everybody is at the same level of technical know-how as you are and until somebody produces a "How to do it" article many members will be unable to participate. However, knowledge is spreading and I am sure that many new members will be logging-on soon. As for the user group using the board for "official" purposes what do you suggest as an official use?. I have not the faintest idea how many Interaks have been sold, the membership is slightly more than 400 so perhaps that means 400+ have been sold. Mind you it is not a requirement of membership to own an Interak so that may not be reliable. Ask David of Greenbank as I am sure we would all like to know. For your further info here is Tom's data. Give him a call, remember he is your new secretary, (grotty legs for a sec though).
"TOMS BULLETIN BOARD" and this is free to all IUGN members.
Data line is 01-573-8822 at 300 BAUD, Speech line is 01-561-2639 for help and advice.)

****

FROM:- MEL SAUNDERS, 7 DRUMCLIFF RD, THUNBY LODGE, LEICESTER, LE5 2LH.
Dear Ed,

It was quite nice to get Interaktion No 9, and thank you for using my Alien Dodge program but you used the old 32 col version. I sent you 64 col versions quite a long time ago, still never mind please find enclosed some more programs.
Note: my sound developer program now has a fine tune routine and can dump register details to the printer!.

On the subject of computers in schools, from what my own 12 & 14 year olds tell me it's mostly game playing (educational I agree) but no connecting it up in the lab, to record events, time, etc.

I am very interested in disk and CP/M but don't know that much about it, especially CP/M why not the newer CP/M plus, can you tell us about CP/M like what utilitys come with it?

Being keen on computer graphics and longing for the day that we get a HI-RES VDU card, I liked Bruce Joyce's routines and it is easy to see where the RETURNS go, but CR seemed to be missing from the INPUT in line 20. (EX5)

I have added a few lines to Pete's tower game to give the number of moves needed and an end comment! (I used odd numbers then I renumbered it):

I've been tempted of late with other computers, with colour, and HI-RES graphics, built in disk drives, 256K ram, and most of all a CP/M machine with microsoft basic and 2x 5.25" drives for £230. But today was a day for real thinking and I don't want a plastic computer no matter how nice!. I want some thing I know about, where all my ram is, whats the rom doing? what I/O ports are being used, something I can abuse a bit, add my own circuits like a real time clock, I/O ports, joystick!

Best wishes to all
Mel Saunders

(ED:- CP/M plus is a slightly enhanced version of CP/M. It can handle banked Ram by using it as a Ram disk. It offers the single user few real gains for a much greater cost outlay. My opinion is to stay with CP/M 2.2 on 3.5" drives until the ZB00 or something really better comes along. Microsoft Basic is available on CP/M 2.2 so when your disks are up you can get a copy. If you

want Hi-RES buy a terminal, you can alter your BIOS to drive it from CP/M and program it independantly. I use the Intelgraph terminal card which gives 80x25 text and 512x256 graphics. But many other cards are available. You could talk to Tom Evans about colour cards as I belive he has implimented one on the Interak. Don't just stick with Greenbank, decide what you wish to add and simply add it on yourself. Thats what it's for - to experiment with.)

****

FROM:- ROY HARRISON, RAMORY, 102 HESTON ROAD, HESTON, MIDDLESEX, TW5 COP.

Dear Ed,

In responce to "Cri de Coeur", newsletter number 7 and trusting that the following is not too simple an exercise, how about the following :-

First some information though. In machine code the commands and data are translated by the operator into hexadecimal characters by using tables, to suit the processor in use, (ours is the Zylog Z80) but to just get the feel for programming in machine code without becoming too involved here are some codes with their meanings, as used for this exercise :-

CD is a call (to a memory location)
3E load the A (accumulator) register
21 load the HL register pair with a screen address.
77 load the address that is being pointed to by the HL register, with the contents of the accumulator
23 increment HL (add one to its contents)
2A is the character "*" or an asterisk (data)

We will use a couple of Zymon's routines by calling them with our CD instruction. One is the Fill command, one is a delay, the other a clear screen. We shall also use a screen address where we shall display our data.

Start by filling the memory area with FF's (traps) by using the command.

F 0800 0900 FF then using Zymon's M command enter the following.
At the prompt type :-

```
M 0800 CD <R>  ..... <R> is Retuern
  0801 C3 <R>  Clear screen routine address low
  0802 06 <R>  Clear screen routine address high
  0803 3E <R>  Load A with ..
  0804 2A <R>  "*"
  0805 21 <R>  Load HL with first screen address ...
  0806 0C <R>  F10C
  0807 F1 <R>
  0808 77 <R>  Write A to the address in HL
  0809 23 <R>  Increment HL

  080A 3E <R>  Load A with ...
  080B 2A <R>  "*"
  080C 77 <R>  Write A to the next screen address, held in HL
  080D 23 <R>  Increment HL
```

Repeat the previous 4 commands until you have enough char's or get fed up but remember the address will increment each time that you press return <R>. (Or just continue from here.)

```
  080E CD <R>
  080F F2 <R>
  0810 04 <R>
  0811 CD <R>
  0812 C3 <R>
  0813 06 <R>
  0814 CD <R>
  0815 F2 <R>
  0816 04 <R>
  0817 C3 <R>
  0818 03 <R>
  0819 08 <R>
  081A ..... This time Press Control and C together, and a new command of
E 0800 <R> should cause the "*" to flash 'on and off' on the screen.
```

You can of course use any data, for instance, to spell out your name, remembering that "A" In Hex is 41, "B" is 42 and so on.

Yours
Roy Harrison

[ED - Thank you Roy for a handy experimental routine which will help quite a lot more people than you realise. I hope you don't mind me including your work as a letter, but I think it reads friendlier than if I transcomposed it into a more formal layout. I wonder If you could apply your knowledge to helping more of the membership by producing a series of articles, working up from square one, each Introducing one new aspect with an example such as you have done here.]

****

FRDM:- R. BARSBY, 24 STANLEY GRDVE, RICHMDND, N.YDRKS, DL10-5AU.
Dear ED,

Does the Newsletter exist? Since joining last year I received a sample copy, then purchased back numbers and have heard nothing since. I have not quite got my Interak system running yet but hope to do so very soon. I have issues 1 to 9 Inc and find them very Interesting. Am I up to date? I have written to P.Vella with SAE but to no avail. Is he still in the country? It is impatient anticipation of trying out some of the ideas in the newsletter which makes me write.

Yours sincerely
R.Barsby.

[Ed - I have written to Mr Barsby, If anyone else Is having this sort of trouble please write in. We will do our best to sort things out. Oh by the way; I tried recently, to phone Pete in Burmuda, but they said he was on his boat and that I should ring back later.]

****

DISK SOFTWARE

The newsletter will list any disk software that members may wish to make availiable to others. It should run under CP/M 2.2. Please note that the supplier is liable for bug repairs, delivery and assistance. The phrase "standard device" means as for normal CP/M BIOS calls. If special BIOS drivers are required they should be specified.

CP/M 2.2

The industry standard disk operating system for the Interak computer. Needs a VOU 2K. Minimum Ram is 20K but runs up to 64k. Customised BIOS by Wolf Schroeder with customised MOVCPM. Supplied with multi diameter Formatter and extra utilities. Ready to Boot up and Go.

Contact Greenbank Electronics for further details.

ZYBASIC 4v7

Runs to CP/M 2.2, 56K ram minimum, Interak standard with a VOU 2K. Runs on any standard BIOS. Supports CP/M disk file program storage. Fully downward compatable with previous versions and will load programs from tape or disk, execute them, and save them to tape or disk. Supplied on a 3.5" disk with run time program source files.

Contact Greenbank Electronics for further details.

****

TAPE SOFTWARE

See CONTACTS page at the end of this issue for "ORDER FROM" addresses.
Software supplied is the responsibility of the "ORDER FROM". Please deal directly with the "ORDER FROM" in the event of bugs ect.
You may use this section to sell software to other users. Send a brief description of your product giving details of its distribution and price, to the EDITOR. Note that you will be responsible for the support of your own product. The newsletter cannot be held responsible for or get involved with duff code or distributers. Of course we will publish letters deriding any product that fails to live up to its claims.

MACHINE CODE

NAME	DESCRIPTION	VOU	ORDER FROM	COST
FIGFORTH	FORTH COMPILER	2K	P.VELLA	£15.00
INTERPLAY	BULLETIN BOARD DRIVER	2K	M&M ELECTRONICS	£ 4.00
MEGABUG	DEBUG/TRAINING PACKAGE	2K	P.VELLA	£13.00
VELTEXT	TEXT EDITOR	2K	P.VELLA	£ 5.00
XTAL BASIC	14K BASIC	2K	P.VELLA	£40.00
ZYBASIC 3A	INTERAK BASIC (TAPE)	2K	GREENBANK	£15.95
ZYBASIC 3C	INTERAK BASIC (ROM)	2K	GREENBANK	£27.75
ZYMON 2.V203	INTERAK MONITOR	2K	GREENBANK	£15.95

XTAL BASIC

NAME	DESCRIPTION	VOU	ORDER FROM	COST
AWARI	GAME	2K	M.SAUNDERS	PP
BIORYTHMS		2K	M.SAUNDERS	PP
CHAR DES	CHARACTER DESIGNER	2K	M.SAUNDERS	£ 5.50
I-SPY	GAME	2K	M.SAUNDERS	PP
SOUND DEV	SOUND DEVELOPMENT	2K	M.SAUNDERS	£ 5.50

Key: PP = Postage & packing. POA = Please enquire (Phone for price.)

FORTHCOMING ATTRACTIONS

The following is a brief list of submissions for publication All the below will be printed in future newsletters. If you can't find yours here please contact me ASAP and I will try to sort it out.
Can I thank all of you for your support and interest in the dissemination of ideas and knowledge about the Interak computer.
This list is not in any publication order, it is to give you some idea of the wealth of skill and knowledge that is still to be published.
May your disks spin in harmony and your code run true - Ed.

Weaver (MC) . Un-named prog ..... Allan C Weaver
Black holes (MC32) ..... Alan C Weaver
Biorhythms, Code and Modulo simulation, Calendar Printer and
Simulating 3D arrays (Z32) ..... Paul Nicklin
Dicepoint (Z32) ..... Pete Vella
Poolspick (Z32) ..... Pete Vella
Dooxxx (Z32) ..... Pete Vella
Count (Z32) ..... Pete Vella
Flyer (Z32) ..... No by line
Mash-64 (Z64) ..... Bob Eldridge
Drawjoy (X32) ..... Mel Saunders
Multest (X32) ..... Mel Saunders
Roulette (X32) ..... Mel Saunders
Fruity (X32) ..... Mel Saunders
Sound-D (X32) ..... Mel Saunders
Memorymx (X32) ..... Mel Saunders
Patterns (X32) ..... Mel Saunders
Acey (X64) ..... Mel Saunders
Chardes (X64) ..... Mel Saunders
I-Spy (X64) ..... Mel Saunders
Multest (X64) ..... Mel Saunders
Sounddev (X64) ..... Mel Saunders
Alien dodge (X64) ..... Mel Saunders
Chars-64 (X64) ..... Mel Saunders
Drawjoy (X64) ..... Mel Saunders
Ispyispy (X64) ..... Mel Saunders
Amazzing (X64) ..... Mel Saunders
Chemist (X64) ..... Mel Saunders
Ladders (X64) ..... Bruce Joyce
Seven eleven (X64) ..... Mel Saunders
Awari (X64) ..... Mel Saunders
Othello (X64) ..... Mel Saunders
Simon (X64) ..... Mel Saunders
Wumpus (X64) ..... Bruce Joyce
Biorythm (X64) ..... Mel Saunders
Memorymx (X64) ..... Mel Saunders
Patterns (X64) ..... Mel Saunders
Fruity (X64) ..... Mel Saunders
Sound-D (X64) ..... Mel Saunders
Assembling Sounds (TEXT) ..... Mel Saunders
Programming exercise (TEXT) ..... Roy Harrison
16 Channel x 16 bit +- CV generator for Analogue music
synthesiser control (TEXT) ..... Alan Payne
Eprom Programming and a design and testing aid (TEXT) ..... David Parkins
Floppy Disk data transfer (TEXT) ..... Simon Waller

****

SMALL ADS

FOR SALE

Tape Labels 3"x0.675" in red, blue, green and yellow	£ 2.00p per 100
Tape/Disk labels 4"x1.5" in blue, green and yellow	£ 3.00p per 100
Disk labels 2.75"x1.5" for 3.5" drives pack of 20x5 colours ..	£ 4.00p per 100
Printout ruler/Template 16"x3.5" smoked grey, with clear lined window for reading printout, 23 flowchart cutouts.....	£ 4.70p each
Disk boxs for 10 x 3.5"	£ 4.50p each
Disk boxs for 10 x 5.25"	£ 6.20p each
Disk binder for 3.5"	£10.40p each
Disk binder for 5.25"	£13.25p each
Disk binder for 8"	£16.60p each
Epsom printer ribbons (80 series)	£ 2.76p each
Epsom printer ribbons (100 series)	£ 4.00p each
Printout binder 9.785"x12" burst/unburst	£ 3.75p each
Printout binder 17.5"x11" burst/unburst	£ 4.25p each

All require postage to be included.
MEL SAUNDERS, 7 DRUMCLIFF RD, THURNBY LODGE, LEICESTER, LE5 2LH.

CONTACTS

CONTACT TAPES. Communicate with other members by cassette tape.
Point Contact tapes,7 Drumcliff Rd, Thurnby Lodge,
Leicester, LE5 2LH.

BACK ISSUES... Can be obtained from:-
D.Parkins, Greenbank Electronics, 92 New Chester road,
New Ferry, Wirral, Merseyside, L62 5AG.

BOOKS..... Lend, borrow, and swop books via :-
R.E.Bowyer, 45 Ford drive, Yarnfield, Stone, Staffs.

BULLETIN BOARD Tom Evans runs "TOMS BULLETIN BOARD" this is free to members.
Data line is 01-573-8822 at 300 BAUD
Speech line is 01-561-2639 for help and advice

BULLETIN BOARD Software and services to the Interak computer.
M & M Electronics, 8 Ayre View, Bride, Isle of man.

DATA SHEETS... Swop, borrow, lend, chip data sheets
7 Drumcliff road,Thurnby Lodge,Leicester,LE5 2LH.

EDITOR..... Send submissions to :-
R.Eldridge, 28 Wycherley Close, Blackheath,London,SE3 7QH.

GREENBANK .... Greenbank Electronics Ltd, 92 New Chester road,
New Ferry, Wirral, Merseyside, L62 5AG.

M&M ELECTRONICS, 8 Ayre view, Bride, Isle of man.

M.SAUNDERS ... M.Saunders, 7 Drumcliff road,Thurnby Lodge,Leicester,LE5 2LH.

MEMBERSHIP.... To join, renew or change your details contact : -
Tom Evans, 129 Cranbourne Waye, Hayes, Middlesex, UB4 0HR.

P.VELLA ..... 19 Ford Drive, Yarnfield, Staffs.

R.ELDRIDGE ... 28 Wycherley Close, Blackheath, London, SE3 7QH.

SUBSCRIPTIONS. For information and payments please contact : -
Tom Evans, 129 Cranbourne Waye, Hayes, Middlesex, UB4 0HR.

INTERAK
User Group
Secretary
TOM EVANS

Data:01-573 8822
Voice:01-561 2639
Taecomm-Interak

Taecomm-Interak

Bulletin Board System and Interak User Group News
Data:01-573 8822 Voice:01-561 2639

129, Cranborne Waye, Hayes, Middlesex. UB4 0HR.

10th August, 1986.

*** STOP PRESS ***

New Interak BB has started testing daily
from the hours of 11pm through to 7am
the following morning.

FLASH GORDON'S BULLETIN BOARD
300 Baud 2 Stop No Parity
Dronfield (0246) 410873
Address:-
The Planet Mars
C/O 229, Stonelow Road,
Dronfield,
Derbyshire.
S18 6ER.
System Operators
Flash Gordon,
Dale Arden,
and Professor Zarkov.

Try it NOW.....

Tom.

Sysop and Secretary: Tom Evans.

On Line Daily 7pm through 7am and all day Sunday
IUGN published quarterly. Editor: BOB ELDRIDGE.